NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

DEVELOPMENT OF A LOCOMOTION INTERFACE FOR PORTABLE VIRTUAL ENVIRONMENT SYSTEMS USING AN INERTIAL/MAGNETIC SENSOR-BASED SYSTEM AND A RANGING MEASUREMENT SYSTEM

by

Chuan Hao Yang

March 2014

Dissertation Supervisor: Xiaoping Yun

This dissertation was performed at the MOVES Institute

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2014	Dissertation

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
DEVELOPMENT OF A LOCOMOTION INTERFACE FOR PORTABLE VIRTUAL ENVIRONMENT SYSTEMS USING AN INERTIAL/MAGNETIC SENSOR-BASED SYSTEM AND A RANGING MEASUREMENT SYSTEM	
6. AUTHOR(S) Chuan Hao Yang	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	

13. ABSTRACT (maximum 200 words)

This dissertation describes the development of an integrated locomotion interface for building self-contained, portable, and immersive virtual environment (VE) systems. Such VE systems do not rely on any infrastructure support and can be used in indoor/outdoor open spaces. The natural walking motions of the user are utilized as a means of signal generation to drive the locomotion interface, which provides the user with a higher sense of presence.

This work investigates the use of two types of measurement systems, the inertial/magnetic measurement units and the ranging measurement systems, to develop a locomotion interface for portable VE systems. Algorithms were developed for each of the two systems to provide the necessary functionalities of the desired locomotion interface. Fusing measurements from a head-mounted and a foot-mounted inertial/magnetic sensor, a locomotion interface was developed for allowing the use of natural walking motions to navigate through virtual environments. To prevent collisions with physical environment boundaries such as walls, a ranging measurement system was used to detect the presence of obstacles. An improved Iterative Closest Point (ICP) algorithm was developed for map-building of the physical environment and for estimating the user's orientation and position within the map. A redirected-walking mechanism was utilized for redirecting the user's walking direction away from boundaries in the physical environment. The two types of measurement systems were integrated to constitute a novel locomotion interface for portable VE systems, and its effectiveness was experimentally tested and demonstrated.

14. SUBJECT TERMS	15. NUMBER OF PAGES
locomotion interface, virtual environment, inertial/magnetic measurement unit, ranging measurement system, map-building, localization, redirected walking, iterative closest point.	171
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DEVELOPMENT OF A LOCOMOTION INTERFACE FOR PORTABLE VIRTUAL ENVIRONMENT SYSTEMS USING AN INERTIAL/MAGNETIC SENSOR-BASED SYSTEM AND A RANGING MEASUREMENT SYSTEM

Chuan Hao Yang
Major, Taiwan Army
B.S., Chung Cheng Institute of Technology, National Defense University, Taiwan, 2001
M.S., Naval Postgraduate School, 2005

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: _____
Chuan Hao Yang

Approved by:

_____	_____
Xiaoping Yun Professor of MOVES Dissertation Supervisor	Eric Bachmann Professor of MOVES
_____	_____
Don Brutzman Professor of MOVES	Peter Chu Professor of MOVES
_____	_____
Christian Darken Professor of MOVES	Roberto Cristi Professor of ECE

Approved by: _____
Christian Darken, Chair, MOVES Academic Committee

Approved by: _____
Peter J. Denning, Chairman, Department of Computer Science

Approved by: _____
Douglas Moses, Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This dissertation describes the development of an integrated locomotion interface for building self-contained, portable, and immersive virtual environment (VE) systems. Such VE systems do not rely on any infrastructure support and can be used in indoor/outdoor open spaces. The natural walking motions of the user are utilized as a means of signal generation to drive the locomotion interface, which provides the user with a higher sense of presence.

This work investigates the use of two types of measurement systems, the inertial/magnetic measurement units and the ranging measurement systems, to develop a locomotion interface for portable VE systems. Algorithms were developed for each of the two systems to provide the necessary functionalities of the desired locomotion interface. Fusing measurements from a head-mounted and a foot-mounted inertial/magnetic sensor, a locomotion interface was developed for allowing the use of natural walking motions to navigate through virtual environments. To prevent collisions with physical environment boundaries such as walls, a ranging measurement system was used to detect the presence of obstacles. An improved Iterative Closest Point (ICP) algorithm was developed for map-building of the physical environment and for estimating the user's orientation and position within the map. A redirected-walking mechanism was utilized for redirecting the user's walking direction away from boundaries in the physical environment. The two types of measurement systems were integrated to constitute a novel locomotion interface for portable VE systems, and its effectiveness was experimentally tested and demonstrated.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
AHRS	attitude heading reference system
API	application programming interface
CAVE	cave automatic virtual environment
FQA	factored quaternion algorithm
FRD	forward-right-down
FRMSD	fractional root mean squared distance
GPS	global positioning system
HIVE	huge immersive virtual environment
HMD	head-mounted display
ICP	iterative closest point
IMMU	inertial/magnetic measurement unit
IMU	inertial measurement unit
MEMS	micro-electro-mechanical systems
ML	maximum likelihood
NED	north-east-down
ODT	omni-directional treadmill
PNS	personal navigation system
PPT	precision position tracker
RMS	root-mean-square
SA	simulated annealing
SLAM	simultaneous localization and map-building
SONAR	sound navigation and ranging
VE	virtual environment
ZUPT	zero-velocity update

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I have received support and encouragement from many individuals over the past four years. I would like to express my deep appreciation to my dissertation supervisor, Professor Xiaoping Yun, for his guidance and support throughout the research as it moved from an idea to a more tangible study, and finally became a completed work. I would also like to thank all my dissertation committee members, Professor Eric Bachmann, Professor Don Brutzman, Professor Peter Chu, Professor Christian Darken, and Professor Roberto Cristi, for all of the valuable advice they provided.

I am grateful for the help and ideas offered by Professor James Calusdian, especially during integrating the system components. His support in coping with the mechanical and electronic hardware issues definitely made the entire research easier.

Last, but not least, I would like to thank my wife, Yihua Peng, for taking great care of me as I buried myself in the research work.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

A virtual environment (VE) is a computer-based simulated environment, in which the users can interact with one another (other users) or with the objects (such as doors, windows, and furniture) contained in this environment. The form and structure of a virtual environment entirely depends on the purpose of constructing it. For example, it can be a museum for educational purposes, a battlefield for military training purposes, or any public place for social communications or entertainment purposes.

An immersive virtual environment is often referred to as a spatially large life-sized virtual environment. The user's awareness of physical self is represented by an avatar (a graphical representation of the user) in first person encompassed by the engrossing scene of the environment. One type of system configuration for achieving the effects of immersion is to adopt a head-mounted display (HMD) with circumaural headphones. This allows the system to be able to effectively block or diminish the user's visual and aural information regarding the actual physical environment and to replace it with that of a designated virtual environment. An example of the head-mounted displays is shown in Figure 1. The user's sense of presence is better enhanced by this means than by using a regular computer screen and speakers. (Presence is defined as the subjective experience of being in one place or environment, even when one is physically situated in another [1]. Examples of other studies regarding the sense of presence can be found in [2], [3] and [4].)

Figure 1. A person wearing a head-mounted display (from [5]).

Another approach to achieving immersion is by adopting a cave automatic virtual environment (CAVE)-like system infrastructure. The CAVE is a room-sized theater that was first developed in the Electronic Visualization Laboratory at the University of Illinois in Chicago, and it was announced and demonstrated at the 1992 SIGGRAPH [6]. The walls and floor of the CAVE are generally made up of rear-projection screens and a down-projection screen, respectively. The projected images on the screens are to be converted into three-dimensional (3D) graphics seen by the user through a pair of special glasses. The system infrastructure also provides 3D sound to complement the graphics. An example of the CAVEs is shown in Figure 2.

Figure 2. A person accesses an immersive virtual environment through the CAVE (from [7]).

There also exist some vehicle simulator designs that reproduce the characteristics of real vehicles in virtual environments which are used effectively in pilot training. An example of this kind of simulators is shown in Figure 3.

Figure 3. Vehicle simulator: a soldier tests out a heavy-wheeled-vehicle driver simulator (from [8]).

Although the development of immersive VE systems has been thriving in recent years, there are still issues that need to be addressed.

VE systems nowadays generally utilize specialized facilities with a sophisticated pre-installed infrastructure, such as the previously mentioned CAVE-like systems and vehicle simulators, or some tracking areas equipped with certain kinds of position tracking systems (e.g., an optical-based tracking system) on the walls of specific laboratory environments. There are also some infrastructures utilizing the treadmill designs [9]–[13] as the user locomotion interfaces, which allow the users to navigate through virtual environments using relatively more natural walking gestures than other designs, such as walking-in-place [14] or leaning [15]. Despite the great performance of those facilities, to use them requires the users to schedule and travel to certain locations. Although these facilities were often built at great expense, they are not as easily accessible to the public as they need to be.

Other popular locomotion interfaces of VEs include keyboards, mice, joysticks, or certain game controllers that provide limited sense of presence to users. To enhance the user's sense of presence, one of the approaches is to increase the requirements of user's body motions when interacting with virtual environments [4]. Many locomotion interfaces were thus developed to utilize motion tracking designs that require actual body gestures from the users. One approach is to utilize the user's walking gestures as a kind of input signal to the locomotion interface. However, some of such interfaces as the walking-in-place or leaning designs were built to accommodate themselves to the limited tracking areas such as the CAVE-like infrastructures and require body gestures that are not as natural as normal walking motions. Some may even cause effects such as simulator sickness.

Motivated to overcome limitations of the existing VE systems, the main goal of this research is to develop a locomotion interface for building self-contained, portable, and immersive VE systems. Such VE systems do not rely on any infrastructure support and can be used in any open indoor or outdoor space. The natural walking motions will be utilized as a means of signal generation for the locomotion interface to provide users

with a higher sense of presence, since these types of motions are exactly the same ones humans use to navigate environments around them in daily life.

B. SIGNIFICANCE AND POTENTIAL IMPACT

In education and training applications, the cost associated with the construction of immersive VE system facilities and the amount of time taken for the trainees to travel to these facilities is huge. Moreover, these facilities often allow only a limited number of people to access the virtual environment systems at a time. They are therefore relatively inaccessible to most people considering the schedule and travel cost. By developing a locomotion interface for building portable immersive VE systems that can function as effectively as these facilities with relatively lower cost, education and training can be made accessible to more people.

Many of the user locomotion interfaces adopted in current immersive VE systems utilize keyboards, mice, joysticks, and other game controllers. Some better designs adopt walking-in-place or leaning, which still provide users with limited sense of presence. The treadmill designs allow the user to navigate through virtual environments using relatively natural walking motions. However, they rely on infrastructures that do not meet the requirement of portability. To achieve a better education or training result, these designs need to be improved. By developing a locomotion interface that allows the user to use the completely natural walking motions that coincide with the real-life conditions to navigate through virtual environments, the education and training efficiency will be greatly enhanced.

C. SPECIFIC OBJECTIVES OF THE RESEARCH

As mentioned earlier, the main goal of this research is to develop a locomotion interface contributing to self-contained, portable, and immersive VE systems. The requirement for a self-contained and portable VE system is to contain all the necessary components (subsystems) that provide complete functionalities without needing support from any pre-installed infrastructure, and that are light enough in weight to be carried by a person. To enhance the sense of presence, the natural walking motions will be utilized as a means of signal generation for the locomotion interface.

A head-mounted display-based system with circumaural headphones is adopted because of its better portability compared with that of a CAVE-like system. The head-mounted display is equipped with an inertial/magnetic measurement unit (IMMU) to determine the orientation of the head. To detect the natural walking motions, additional IMMUs are mounted on shoes to measure the accelerations and rotation rates of the feet. A mechanism will be established to relate these real-time measurements (from both the IMMUs on the head and the feet) to the walking motions in the virtual environment.

Such a VE system will be characterized by its locomotion interface that utilizes natural walking motions, and it will work if the user is in an infinitely large open space. However, in an actual open space with limited size, the user may collide with obstacles (walls or other objects) as he/she navigates through the VEs. Thus, there is a need to establish a mechanism for the user to avoid obstacles. To achieve this purpose, certain ranging measurement systems will be integrated into the design of locomotion interface to measure the distances and directions of obstacles relative to the location of user.

To summarize, the specific objectives of this research are:

- To investigate the feasibility of using two types of measurement systems, namely, inertial/magnetic measurement units and ranging measurement systems, as a means to build a locomotion interface for self-contained, portable, and immersive VE systems.
- To develop algorithms for each of the two types of measurement systems as candidates for the necessary functionalities of the locomotion interface.
- To integrate the two types of measurement systems and their algorithms to develop a desirable locomotion interface.

D. SYSTEM HARDWARE CONFIGURATION

The hardware components adopted in this research for investigating the development of a locomotion interface will be introduced in this section. Note that the research to be conducted will not be hardware specific, and that all of the components may be replaced by choice as long as the necessary functionalities can be provided.

The head-mounted display along with a video control unit adopted in this research is an nVisor SX60 by NVIS Inc. [16]. The portable immersive VE system consisting of

such a head-mounted display system and a laptop computer for running the virtual environment is shown in Figure 4.

The inertial/magnetic measurement unit attached to the head-mounted display is an InertiaCube2+ by InterSense Inc. [17] as shown in Figure 5. The inertial/magnetic measurement unit attached to the shoe is a 3DM-GX3-25 by LORD MicroStrain Inc. [18] as shown in Figure 6.

Figure 4. The portable immersive VE system utilizing an nVisor SX60 head-mounted display system. The figure on the left shows the head-mounted display and the backpack equipped with the video control unit and a laptop computer for running the virtual environments. The figure on the right shows a person carrying a similar VE system.

Figure 5. The InertiaCube2+ inertial/magnetic measurement unit (from [17]).

Figure 6. The MicroStrain 3DM-GX3-25 inertial/magnetic measurement unit (from [18]).

An ultrasonic ranging measurement system was initially investigated because of its low cost and simplicity. An example configuration of such a system is shown in Figure 7. The system contains an ultrasonic control unit and eight ultrasonic transducers arranged in two sets of semi-rings around the user.

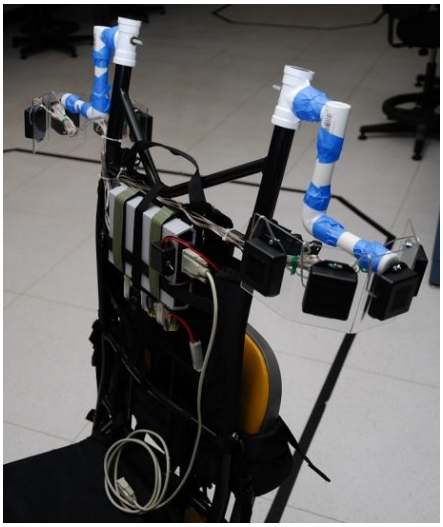


Figure 7. An ultrasonic ranging measurement system. The same backpack as shown in Figure 4 is used but only equipped the ultrasonic ranging measurement system. The system contains an ultrasonic control unit and eight ultrasonic transducers arranged in two sets of semi-rings around the user.

The ultrasonic ranging measurement system is replaced due to its low resolution (eight measurements over a 360-degree coverage area) and short effective range (under 7

meters). The replacement is a UTM-30LX laser ranging measurement system by Hokuyo Automatic Co., Ltd. [19]. Such a laser-based ranging measurement system has a higher resolution (1081 measurements over a 270-degree coverage area) and a longer effective range (30 meters). Furthermore, it operates more efficiently than the ultrasonic ranging measurement system, since the light travels much faster than the sound. The UTM-30LX ranging measurement system is shown in Figure 8. Its integration into the portable immersive VE system is shown in Figure 9. It is intended to be placed above the head to prevent the laser beams from being blocked by the human body.

Figure 8. The UTM-30LX ranging measurement system (from [19]).

Figure 9. The UTM-30LX ranging measurement system integrated into the portable immersive VE system as shown in Figure 4.

E. DISSERTATION OUTLINE

The goal of this dissertation is to develop a locomotion interface contributing to self-contained, portable, and immersive VE systems. Some background knowledge that serves as the basis of this dissertation is provided in Chapter II. In search of a mechanism that describes the physical environment surrounding the user for the subsequent obstacle avoidance task, the ranging measurement system-based map-building and localization algorithms are investigated and developed in Chapter III. The improved map-building and localization algorithms that can be utilized in particular types of environments are presented in Chapter IV. Without regard to the obstacle avoidance task, the inertial/magnetic sensor-based locomotion interface for virtual environment systems is developed in Chapter V. To approach the obstacle avoidance task, the locomotion interface developed in Chapter V is integrated with a ranging measurement system. Such an integrated locomotion interface is presented in Chapter VI. Lastly, the conclusion and recommendations for future researches are summarized in Chapter VII.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND KNOWLEDGE

In this chapter, several techniques are briefly introduced to aid the reader in understanding the construction of a new locomotion interface for self-contained, portable, and immersive virtual environment systems. The first part of this chapter includes discussion of several sensor techniques. The ultrasonic and laser sensors are considered as candidates to collect information of the physical environment for avoiding collisions with obstacles, while the inertial/magnetic sensors are adopted to measure the user's body motions for building the major portion of the desired locomotion interface. An alternative approach for avoiding obstacles is to keep track of the user's current location in the physical environment. The inertial/magnetic sensor-based personal localization and the simultaneous localization and map-building are considered as candidates to serve this purpose. Finally, the redirected-walking technique is introduced to utilize the knowledge of user's current location in the physical environment to avoid collisions with obstacles.

A. SENSOR TECHNIQUES

1. Ultrasonic Sensors

Range finding is a common use of ultrasound and is also called SONAR (sound navigation and ranging). Ultrasonic ranging devices have been widely used in robotics. Some are used to track environmental features for underwater navigation [20], [21]. Most of other applications fall into the subjects of obstacle avoidance [22], [23] and target tracking [23] depending on the configurations (types and position arrangements) of these devices. However, the main function of the ultrasonic ranging devices is simply distance estimation.

There are two types of ultrasonic ranging devices based on the means by which distance estimation is performed. One type of these devices includes at least an ultrasonic transmitter and an ultrasonic receiver. The distance being estimated is that between the transmitter and receiver, and the estimation is accomplished by measuring the time taken for the ultrasonic wave to travel from the transmitter to the receiver, which is only a one-way trip (as shown in Figure 10). The other type uses only one ultrasonic transducer that acts as the transmitter and also the receiver. The distance being estimated is that between

the ultrasonic transducer and the nearest obstacle (if any) within the maximum detection range. This distance is estimated by measuring the time taken between the transmitting of the ultrasonic wave and the receiving of its echo, which is a round trip (as shown in Figure 11).

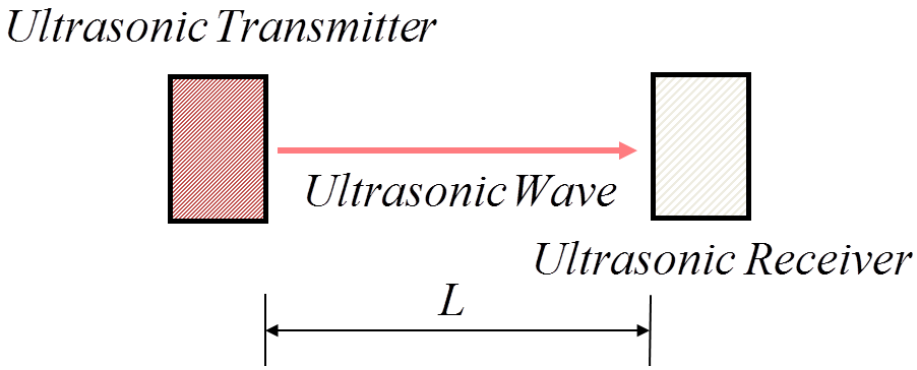


Figure 10. Notional distance estimation process of a transmitter/receiver type of ultrasonic ranging devices.

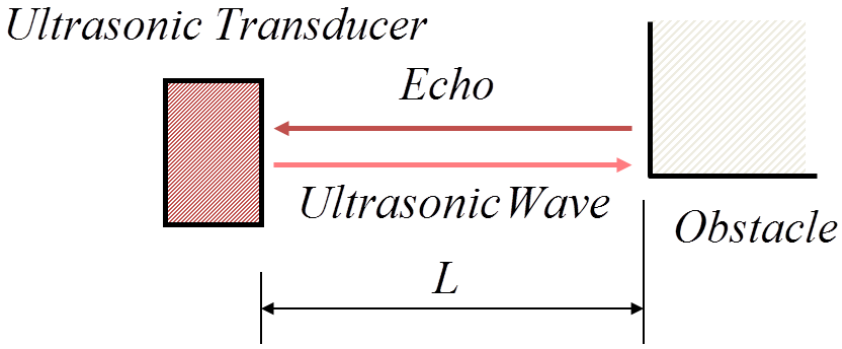


Figure 11. Notional distance estimation process of an ultrasonic transducer.

The transmitter/receiver type of ultrasonic ranging devices usually have larger detection ranges than the transducer type because the ultrasonic waves of the

transmitter/receiver type of ranging devices only need to travel one way and thus last farther before attenuating below a certain detection threshold. The transmitter/receiver type of ultrasonic ranging devices are therefore relatively more suitable for use in target tracking, while the transducer type are more suitable for use in obstacles detection.

An example configuration of a target tracking system using the transmitter/receiver type of ultrasonic ranging devices is shown in Figure 12. The ultrasonic transmitter is attached to the target being tracked and periodically transmits ultrasonic waves to be received by the two ultrasonic receivers. The two ultrasonic receivers are installed at certain fixed locations in the tracking environment, in which the estimated target position is represented in a fixed global coordinate system. They can also be installed on a mobile robot, and the estimated target position will instead be represented in a moving body coordinate system of the mobile robot. The distance D^* and bearing γ of the target relative to the coordinate origin can be estimated as follows [23].

$$D^* = \sqrt{\frac{2D_1^2 + 2D_2^2 - d^2}{4}}. \tag{2.1}$$

$$\gamma = 90^\circ - \cos^{-1}\left(\frac{D_1^2 - D_2^2}{d\sqrt{2D_1^2 + 2D_2^2 - d^2}} \right). \tag{2.2}$$

D_1 and D_2 are the measured distances between the ultrasonic transmitter and the receivers A and B, respectively. d is the distance between the two receivers.

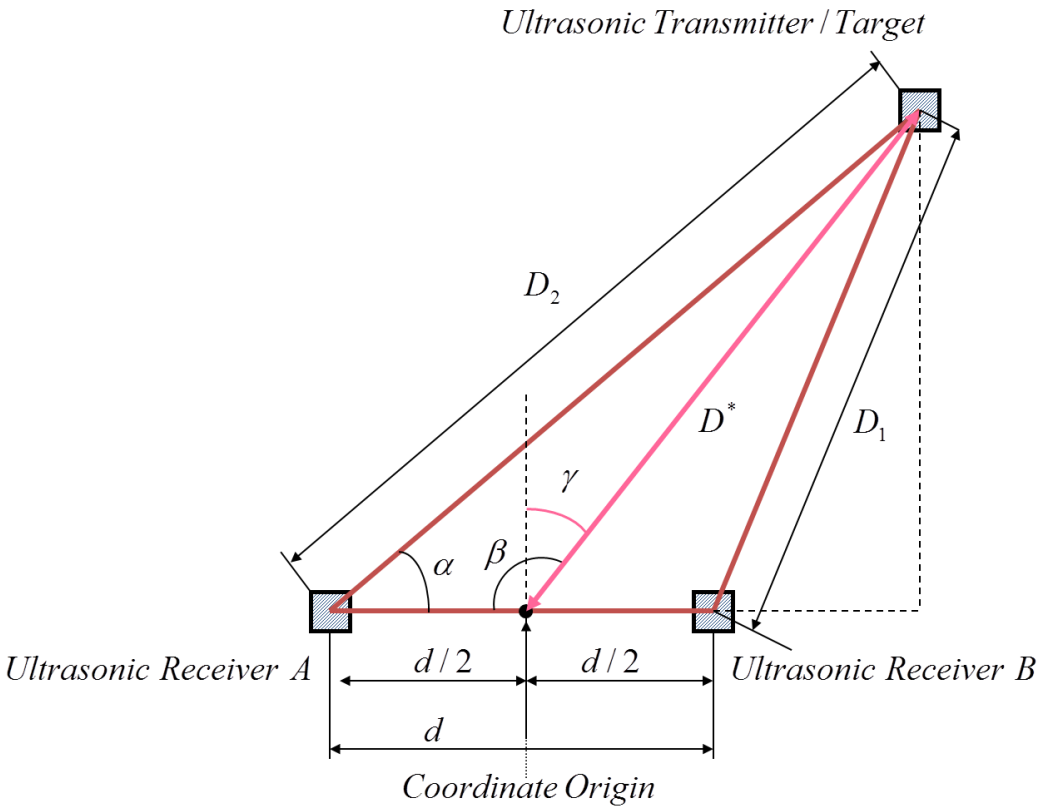


Figure 12. Notional ultrasonic target tracking system (from [23]).

The transducer type of ultrasonic devices are not suitable for use in target tracking because they are not capable of distinguishing between different objects. However, these devices perform fairly well in obstacle avoidance. The ultrasonic ranging measurement system initially built in this research adopts this type of ultrasonic devices. By determining a suitable configuration of a number of ultrasonic transducers, the distance between each ultrasonic transducer and its nearest obstacle can be estimated. The estimated distance and the aiming angle of the transducer make it possible to obtain an approximation of how far and in what direction the obstacles are. The top view of an example configuration of ultrasonic transducers for the ultrasonic ranging system is shown in Figure 13, which is in correspondence with Figure 7. By detecting and estimating the distance from the nearby obstacles (such as walls), the ultrasonic ranging measurement system is able to provide the approximate obstacle information in the local coordinate system of the user. The subsequent obstacle avoidance mechanism may be implemented by using this information.

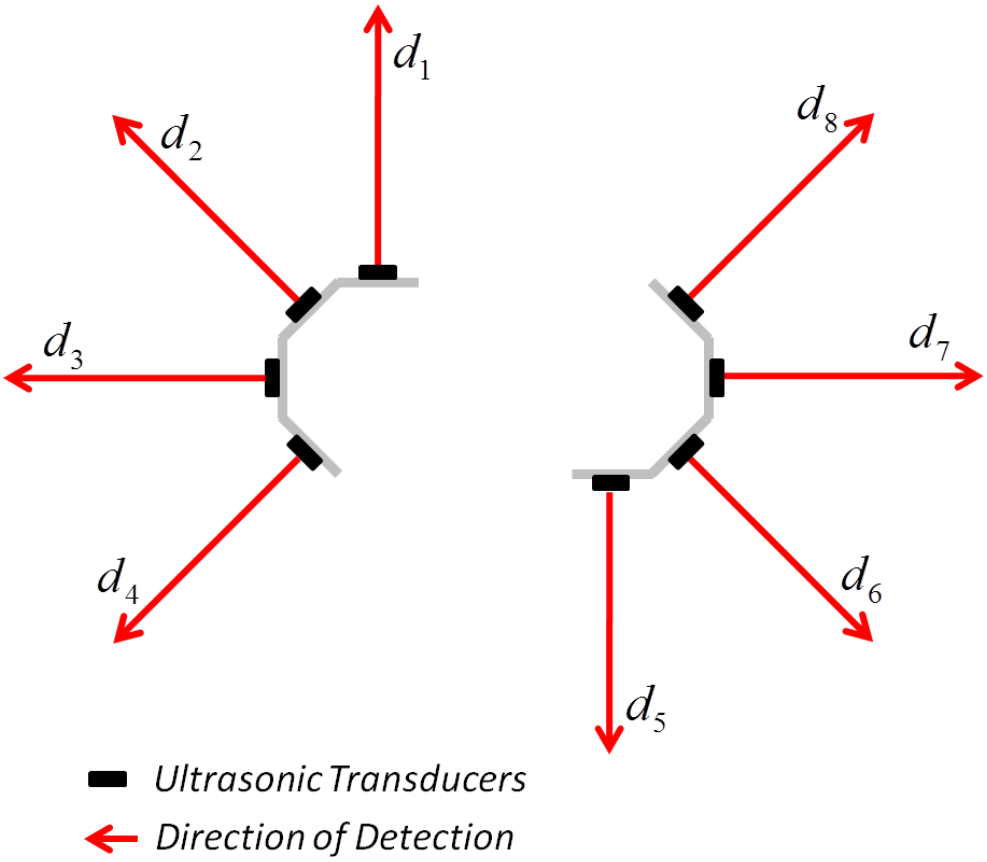
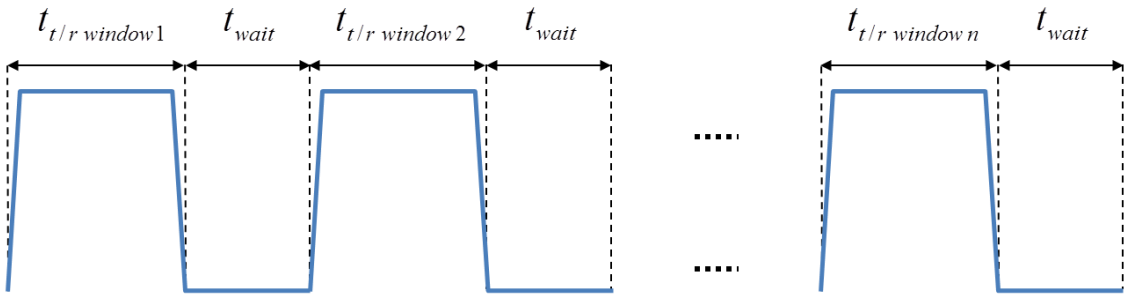


Figure 13. An example configuration of ultrasonic transducers for the ultrasonic ranging measurement system.

To prevent the ultrasonic waves of the transducers from interfering with one another, the transducers are configured to work in sequence, rather than simultaneously. There should also be a blanking period between any two processing periods of transducers for the ultrasonic wave of the previous transducer to attenuate below the detection threshold of the next transducer. This will ensure that every transducer receives its own ultrasonic signal. The only issue with this method is that it takes longer for all the transducers to complete their cycles once. Therefore, the distance estimation update rate may be greatly reduced when the number of transducers used increases. One cycle of the process timing diagram of the ultrasonic ranging system with *n* transducers is shown in Figure 14.



$t_{t/r\ window\ i}$ = *Time allowed for receiving the echo after the ultrasonic wave from the ith transducer is transmitted.*

t_{wait} = *Time allowed for the ultrasonic wave from the previous transducer to attenuate.*

Figure 14. Process timing diagram of the ultrasonic ranging measurement system.

2. Laser-Based Ranging Sensors

One of the techniques laser-based ranging sensors use to determine the distance to an object is the implementation of time-of-flight principle similar to that utilized by the ultrasonic transducers. The sensor sends a laser pulse towards the object and measures the time taken for the pulse to be reflected off the object and returned to the sensor. However, it is difficult to measure the time accurately due to the high speed of light.

In order to enhance the accuracy of determining the time delay between the transmitting and receiving the laser pulse, another technique is based on the calculation of the phase difference between the sinusoidal laser signals (the original and the reflected). The time delay is then indirectly determined. By the phase calculation, it is possible to obtain stable measurements with minimum influence from the object's color and reflectance [24]. This is the same technique utilized by the laser ranging measurement system adopted in this dissertation.

The laser ranging measurement system (UTM-30LX) adopted in this research uses the infrared laser with a wavelength of 905 nanometers to scan a 270-degree semicircular field in a counter-clockwise direction. A notional representation of the scanning area is shown in Figure 15. The guaranteed range is within 0.1 and 30 meters (up to 60 meters with degraded performance). The system measures the distance between

itself and the objects within this range for each of total 1080 angular steps (1081 measurements) that cover the 270-degree field [25]. By this means, the information regarding the environment around the laser ranging measurement system can be obtained.

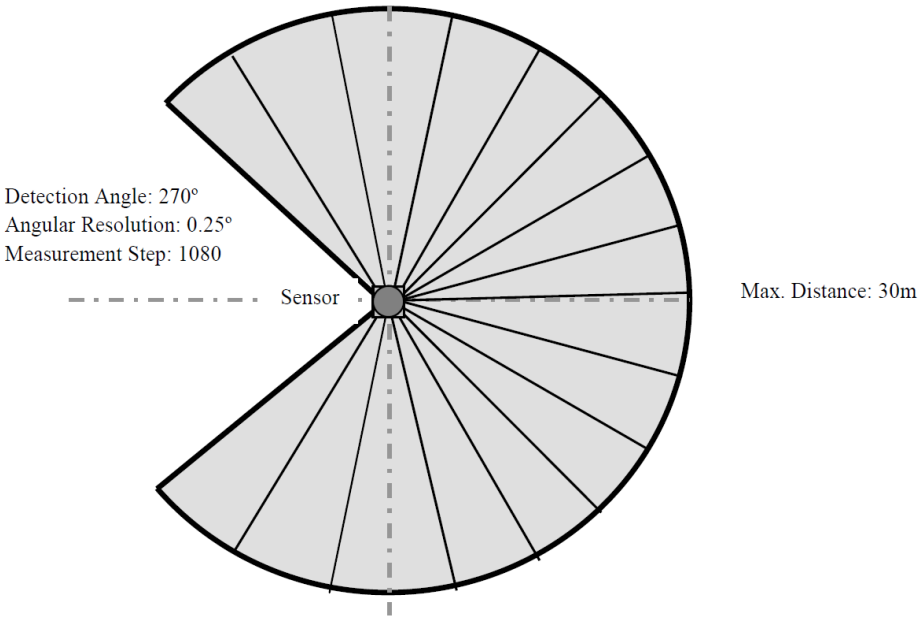


Figure 15. Top view of the notional representation of the scanning area of the UTM-30LX laser ranging measurement system (from [25]).

According to the specifications of such a system [25], the time taken for completing the scanning process once is approximately 25 milliseconds (24 ms for scanning and 1 ms for synchronizing the scan data) not including that for transmitting and processing the data. This is much shorter than the time taken for the ultrasonic ranging measurement system to complete the scanning process once (in seconds, depending on the number of sensor units, and on the distances between the sensor units and the objects). This is one of the advantages of the laser-based ranging sensors which makes them superior to the ultrasonic-based ranging sensors. The laser-based ranging sensors generally have higher resolutions and longer effective ranges as well.

3. Inertial/Magnetic Sensors

The inertial/magnetic sensors (often introduced as the inertial/magnetic measurement units, IMMUs), such as the 3DM-GX3-25 and the InertiaCube2+ used in

this research, are low-cost, high-performance, miniature attitude heading reference systems (AHRSs) utilizing micro-electro-mechanical systems (MEMS) sensor technology [26], [27]. Such an IMMU is comprised of triads of orthogonally mounted accelerometers, angular rate sensors, and magnetometers used to estimate the attitude (position and orientation) of the moving body to which the IMMU is rigidly attached. It is often described as a strap-down measurement unit because of this feature.

The strap-down IMMU provides the raw data including the accelerations, angular rates, and magnetic fields with respect to the IMMU's body coordinate system. It also provides the orientation represented in either the Euler angles (the roll, pitch, and yaw angles) or a quaternion computed by the manufacturer's proprietary algorithm. The strap-down IMMUs are widely used in various fields. For example, they may be utilized for the attitude estimation and stabilization of certain platforms in the field of robotics. They may also be used as the inertial aiding of the global positioning system (GPS) navigation of vehicles.

One of the other applications is the personal localization which is briefly discussed in the next section. This application is especially useful when tracking a person's position and orientation in the indoor environment or when the GPS signal reception is not available.

Another application is the gesture recognition that can be used to build a locomotion interface for the virtual environment systems. By attaching the strap-down IMMUs to the user's head and foot, the user is able to navigate through the virtual environment by using the natural walking motions. The construction of such a locomotion interface is introduced in Chapter V.

B. INERTIAL/MAGNETIC SENSOR-BASED PERSONAL LOCALIZATION

One of the approaches to accomplishing personal localization is through processing the data produced by the IMMUs attached to the person being tracked during motion. A personal navigation system (PNS) consisting of such IMMUs and a computer system for processing the sensor data can be constructed to continuously track the

person’s position and orientation. The system may also keep a complete history of the person’s change in position and orientation over time.

The concept of inertial/magnetic sensor-based personal localization is different from that of automotive navigation that utilizes a GPS. The PNS aims at being self-contained in the sense that it is able to perform all the functionalities without needing support from any other system or infrastructure. The desired capability of such a PNS is to precisely estimate the attitude of the body segment of the person (or user) to which the IMMUs are attached. For example, by attaching the IMMU to the user’s foot, the foot attitude during walking motion can be determined.

The PNS using the strap-down IMMUs forms a kind of relative navigation system where the new position and orientation estimates are computed with respect to the previous ones. One of the advantages of this type of navigation system is that it does not require the use of any infrastructure. However, the disadvantage is that the IMMUs are susceptible to drift errors. The errors tend to grow without bound, which becomes the primary limitation of this kind of navigation system. The estimates obtained by these devices need to be further processed by some means of error correction.

The IMMUs generally used in the PNS provide the raw data from their accelerometers, angular rate sensors, and magnetometers along with the orientation represented in either the Euler angles (the roll, pitch, and yaw angles) or a quaternion computed by the manufacturer’s proprietary algorithm. The acceleration measurements of the IMMUs are represented in their moving body coordinate systems (or body reference frames). Before these measurements can be effectively utilized, they need to be transformed and represented in the fixed global coordinate system (or navigation reference frame) by utilizing the orientation provided by the IMMUs. After the transformation, the velocity and position estimates of the IMMUs can be determined simply by integration and double-integration of the acceleration measurements, respectively. However, as mentioned earlier, the drift errors in the acceleration measurements need to be corrected by some means. Otherwise, the errors will be accumulated into the subsequent estimation process of velocity and position.

There exists a certain point in the gait cycle where a periodic error correction can be conducted. A gait cycle consists of a stance phase and a swing phase. The stance phase of either foot represents the period when this foot is in contact with the ground, while the swing phase represents the period otherwise. A gait phase detection mechanism [28]–[30] is able to distinguish between the stance phase and the swing phase in the gait cycle by comparing the angular velocity of the foot with a specified threshold. A means to correct the drift errors in the acceleration estimates is the zero-velocity update (ZUPT) mechanism used in [28], [29] and [30], which is based on the fact that at the end of swing phase and in the stance phase, the foot comes to rest on the ground and has a zero velocity. By incorporating the gait phase detection mechanism to identify this point in the velocity profile, there is an opportunity to correct the errors accumulated during the integration of the acceleration estimates. An example of the effect provided by the zero-velocity update mechanism is shown in Figure 16.

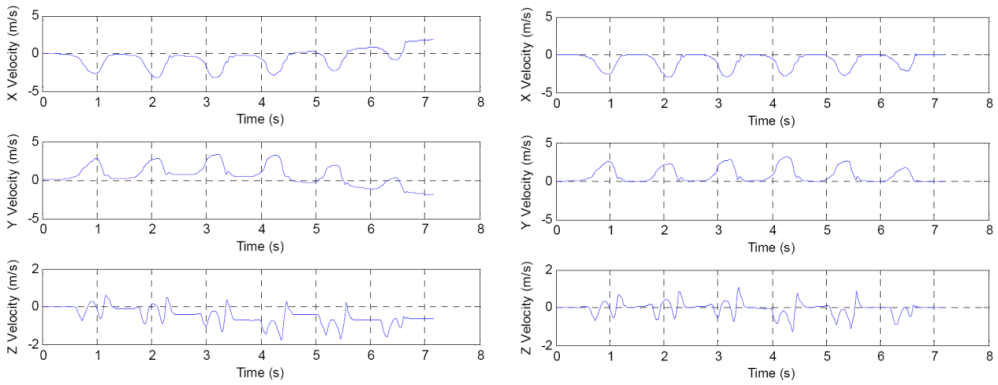


Figure 16. Three components of the velocity (from [28]): velocities obtained by integrating the original acceleration measurements (left) and obtained by integrating the drift-corrected accelerations (right).

The strap-down navigation algorithm combined with the zero velocity update mechanism and the gait phase detection mechanism is shown in Figure 17. The term $L_q(\vec{a}^b)$ is the operator that transforms the acceleration $\vec{a}^b$ in the body reference frame into the acceleration $\vec{a}^n$ in the navigation reference frame by using the quaternion q. The acceleration measurement $\vec{a}^n$ contains gravity $\vec{g}$ that should be removed to determine the actual acceleration. The actual acceleration is then integrated to yield velocity $\vec{v}^n$ given

the initial velocity $\vec{v}_0^n$. By comparing the angular velocity $\vec{\omega}^b$ with a specific threshold, the gait phase detection mechanism detects the stance phase and triggers the zero velocity update mechanism to correct the velocity. The position $\vec{p}^n$ is obtained by integrating the corrected velocity given the initial position $\vec{p}_0^n$.

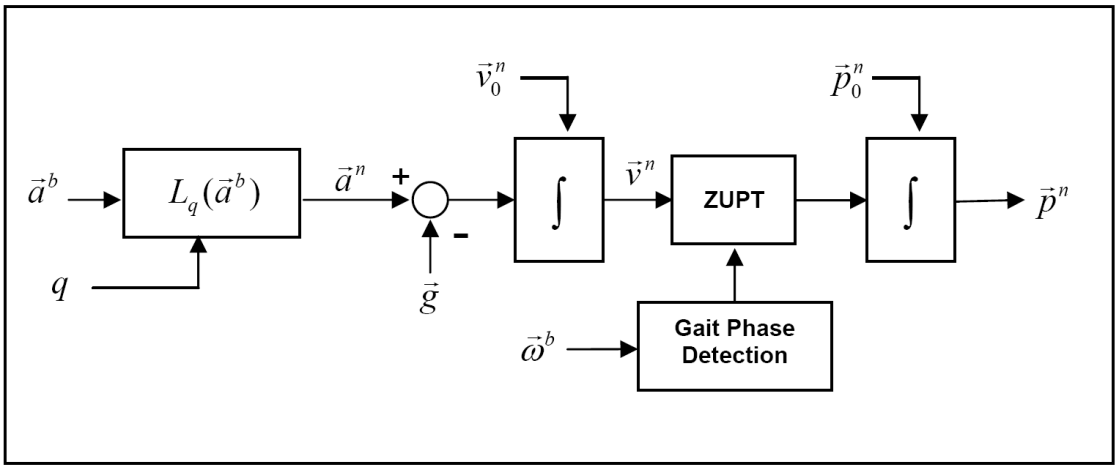


Figure 17. Strap-down navigation algorithm combined with the zero velocity update (ZUPT) mechanism and the gait phase detection mechanism (from [29]).

There exist errors in the orientation provided by the IMMU manufacturer's proprietary algorithm. Since the algorithm is not accessible, it is difficult to cope with the error issue directly. Another means to obtain more precise orientation information is needed.

The factored quaternion algorithm (FQA) developed in [31] is suitable for use in determining the orientation in terms of quaternions for static or slow-moving objects. It computes three distinct quaternions, each of which is corresponding to the rotation about one of the principal axes of the navigation reference frame. The final step of the FQA is to multiply the three principal quaternions to yield a composite quaternion shown as the following.

$$q = q_{Yaw} q_{Pitch} q_{Roll}. \tag{2.3}$$

In the FQA, the roll and pitch quaternions are computed from the acceleration vector measurements, while the yaw quaternion is computed from the magnetic field vector measurements.

A quaternion-based complementary filter developed in [29] and [30] combines the quaternions computed from the FQA (static branch) with those computed using the angular rates (dynamic branch). The complementary filter is shown in Figure 18.

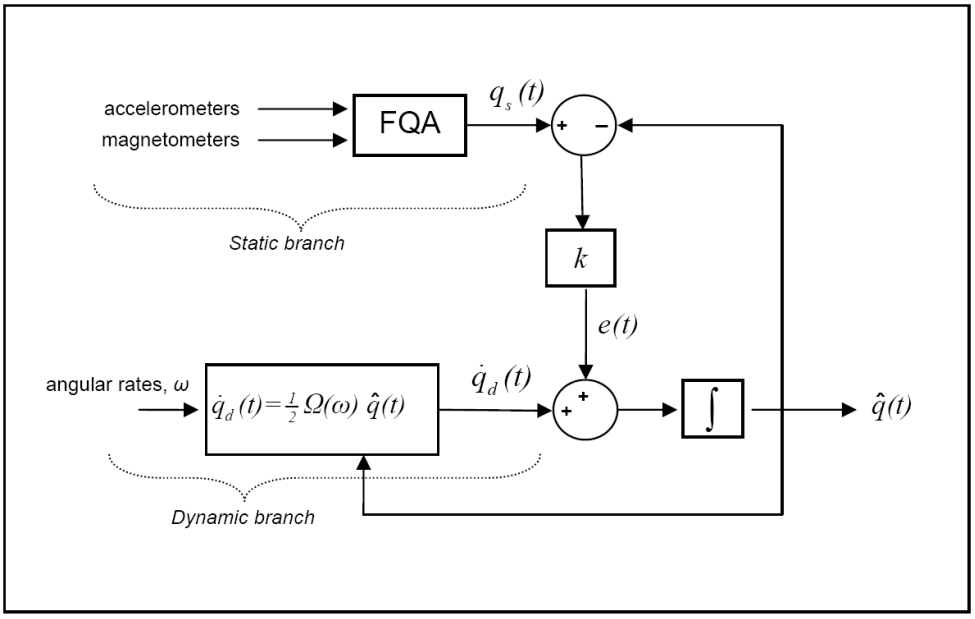


Figure 18. Quaternion-based complementary filter (from [29]).

In the quaternion-based complementary filter shown in Figure 18, the dynamic equation that relates the quaternion estimate $\hat{q}(t)$ to the quaternion rate $\dot{q}_d(t)$ is shown as the following.

$$\dot{q}_d(t) = \frac{1}{2}\Omega(\omega)\hat{q}(t). \tag{2.4}$$

where $\Omega(\omega)$ is a skew-symmetric matrix comprised of the angular rate measurements.

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \tag{2.5}$$

The filter gain k is to adjust the weight of either branch used in the final computation of quaternion estimate $\hat{q}(t)$. The term $e(t)$ is k times the difference between the recent quaternion $q_s(t)$ computed from the FQA and the quaternion estimate $\hat{q}(t)$ from the previous iteration.

The filter gain k plays an important role of the complementary filter. As the gain is decreased, the output will be weighed more by the quaternion derived from the dynamic branch. As the gain is increased, it will be weighed more by the quaternion from the static branch (FQA) instead.

An adaptive gain quaternion-based complementary filter developed in [29] and [30] can adaptively switch the value of complementary filter gain to take advantage of the nature of the gait phase. That is, in order to obtain better quaternion estimates, the gain can be switched to a higher value during the stance phase, and switched to a lower value during the swing phase. This takes into consideration that the foot motion is relatively static during the stance phase and relatively dynamic during the swing phase.

By combining the strap-down navigation algorithm with the zero velocity update mechanism, the gait phase detection mechanism, and the adaptive gain quaternion-based complementary filter, a self-contained inertial/magnetic sensor-based PNS [29], [30] was constructed as shown in Figure 19.

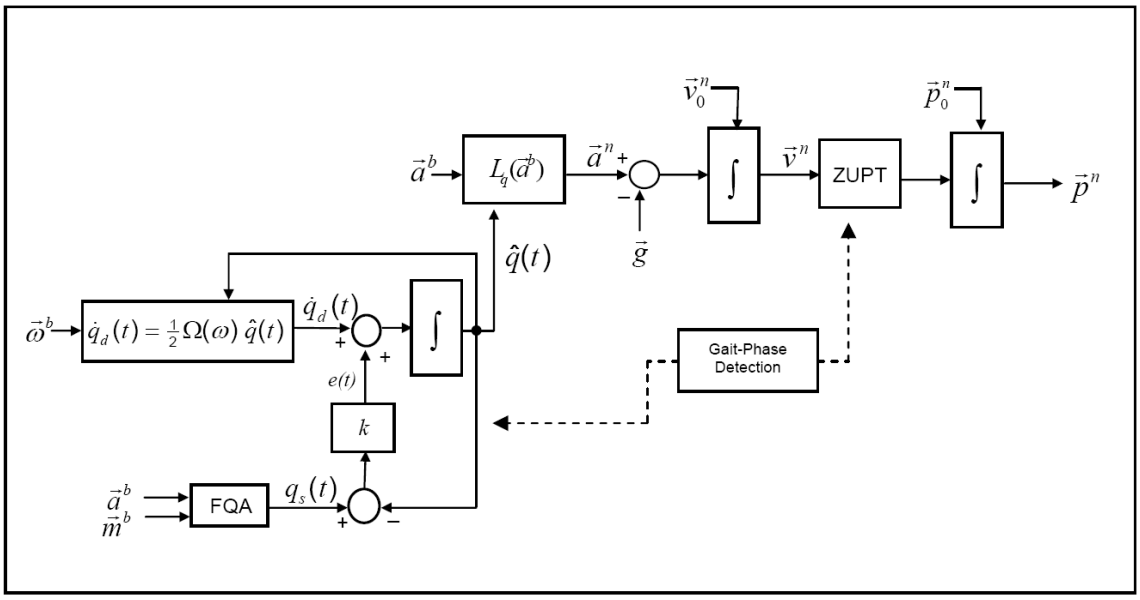


Figure 19. A self-contained inertial/magnetic sensor-based PNS consisting of the strap-down navigation algorithm, the gait-phase detection mechanism, and the adaptive gain quaternion-based complementary filter (from [29]).

C. SIMULTANEOUS LOCALIZATION AND MAP-BUILDING

Simultaneous localization and map-building (SLAM) is one of the techniques used in the field of autonomous mobile robot research. It allows a robot to incrementally build a consistent map of the physical environment where it operates while simultaneously estimating its pose (in terms of its position and orientation) within this map [32]–[35]. The single map to be built is often called the global map, as opposed to the several local maps constructed at different locations.

For a robot to build a local map, it uses the onboard sensors (such as ultrasonic, laser, or other sensors) to scan, collect, and process the information from its surrounding environment relative to its current location. The local maps are often referred to as the local scans, which may be represented by any format, but it must be consistent in illustrating the robot's surrounding environment at all of the locations where the scans are taken. For example, point or line-based representations are adopted for particular uses in this dissertation.

The general concept of SLAM is fairly straightforward. Supposing the transformation (i.e., the relative rotation and position) of the robot between taking two scans is known, the map-building process is completed by applying the transformation to one of the scans and thus allowing this scan to be matched with the other. In other words, the transformation that relates one robot pose with the other is the very same transformation for one scan to be matched with the other. If the localization problem can be solved, so can the map-building problem. However, the robot's pose is often unknown or not accurate due to the errors accumulated in the odometry readings. Under this circumstance, the localization problem may not be solved beforehand, and the determination of transformation needs to be approached from the other aspect through solving the map-building problem (i.e., how the scans are matched). This approach requires determining the correlation between the two scans. Because the two separate problems are highly coupled, they are often treated as a single one—the SLAM study.

The procedure following the match of two scans in the map-building process is to merge them into a single scan which serves as a better representation of the robot's environment than each original local scan. Therefore, as the robot keeps taking scans

along its way, the robot's trajectory and the global map of the environment can be built incrementally.

Regardless how the two matched scans merge together, the SLAM study in this context can be simply referred to as the study of scan matching. However, while the SLAM concept is simple, to cope with several issues related to the scan-matching process is more complicated. For example, current algorithms in the literature that determine the transformation between two scans are innately difficult and environment dependent. The algorithm suitable for a certain type of environment is not necessarily suitable for another. Furthermore, there is an accuracy-efficiency trade-off for almost any existing algorithm, especially for those algorithms that operate in real time. This involves a sacrifice of scan-matching accuracy for higher efficiency or the other way around, depending on different requirements and situations. Several examples of researches seeking improvements in either accuracy or efficiency, as well as in dealing with different types of environments, are introduced.

In [36] and [37], the local scans are collections of line segments obtained from the two-dimensional (2D) laser range measurements. The final 2D global geometric map is generated by the post-processing of the local segment-based scans. The method of integrating two scans is exclusively based on the geometrical information contained in the scans to be integrated. In particular, the angles between the pairs of line segments are considered as a sort of "geometrical landmarks" on which the process is based. In this way, two angles are considered equal when the angle difference is less than a certain threshold value. Similarly, two points are considered to coincide when the distance between the two points is less than some specified threshold value.

In [38], a sonar-based map-building and navigation system was developed for an autonomous mobile robot to operate in unknown and unstructured environments. The system uses sonar range data to build a multileveled description of the robot's surroundings, where the sonar range data are interpreted using probability profiles to determine empty and occupied areas. The local maps that consist of the surrounding occupancy information from multiple points of view are then integrated into a global sensor-level sonar map used for path planning and navigation.

In [39], a sensor model using the probabilistic approach was proposed to translate range data into grid status information of the local map. The Bayes' theorem is used to update the global map. As for the pose of mobile robot, its estimation is conducted by using the incremental maximum likelihood (ML) scan matching. Both the obstacle detection and the mapping accuracy were reported to be improved by fusing the range data obtained from the laser and sonar sensors.

In [40], a fast nonlinear map optimization algorithm was developed to optimize a map even when the initial estimate of the map is poor. Their approach uses a variant of Stochastic Gradient Descent on an alternative state-space representation that has good stability and computational properties.

In dealing with different types of environments (e.g., indoor and outdoor environments), an adaptive perception system was developed in [41] to determine the classification of indoor/outdoor operational environments before applying corresponding systems to implement the SLAM tasks. The operational environment is determined by using the image classification techniques, by which the image features are extracted from video imagery and used to train a classification function using supervised learning techniques. A terrain map that utilized the GPS and inertial measurement unit (IMU) data is used when operating outdoors, while a 2D laser-based SLAM technique is used when operating indoors. The global map is generated by first transforming the indoor local map data into the global reference frame and then combining the transformed indoor local map data with the outdoor map data.

D. REDIRECTED WALKING

In the applications of virtual environment, one of the main efforts is to provide the user a higher sense of presence. This can often be achieved by enhancing the graphic and sound effects. However, this kind of enhancement offers limited improvement if the user's physical body is always stationary. To better resolve this issue, another approach is through building a locomotion interface that allows the user to use real walking motions to navigate through the virtual environment.

Assuming such a locomotion interface is available, there are other issues needing to be addressed. One of the main issues is that the size of the virtual environment is limited to the size of the real physical environment, since the user can never walk beyond the boundaries (walls) of the real physical environment.

Redirected walking is an interactive locomotion technique that captures the benefits of real walking while extending the possible size of the virtual environments [42]–[44].

The way the redirected-walking mechanism works is to rotate the entire virtual environment about the vertical axis of user's viewpoint, which indirectly causes the user to make adjustments in his/her heading direction accordingly in order to follow his/her path to a certain target location in the virtual environment (assuming that the user has a path to a certain target location in mind). This kind of scene-rotation-induced heading direction adjustment prevents the user from walking into obstacles (such as walls) or out of tracking areas in the real physical environment. Ideally, the rotations injected by the redirected-walking mechanism may not be perceived by the user.

The scene rotation can be injected without being detected in several kinds of conditions. These conditions are classified as follows: (1) while the user is standing still, (2) while the user is turning his/her head or body, and (3) while the user is walking [42]–[44].

While the user is standing still, a small baseline virtual environment rotation rate can be injected and will be either unperceived or perceived as self-motion of the user, such as turning the head or rolling the eyes that is often done unintentionally in real life. According to [44], an experimentally determined detection threshold of the baseline rotation rate is 1.0 deg/sec, which can be correctly detected (the presence and direction of rotation) by general users 75% of the time. While the user is turning his/her head or body or while walking, the virtual environment can be rotated even faster so as to steer the user away from obstacles imperceptibly.

A desired result of the redirected-walking mechanism is shown in Figure 20. The three hypothetical paths that the user could take in the virtual environment are actually to be steered toward the same target location in the real physical environment. The block diagram of the redirected-walking mechanism that produces this result is shown in Figure 21.

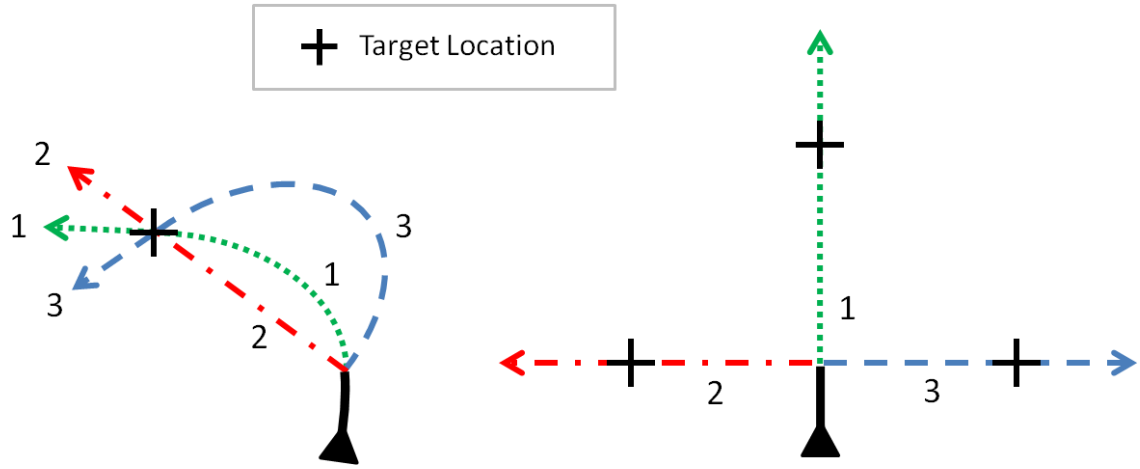


Figure 20. Desired result of the redirected-walking mechanism. Three hypothetical paths that the user takes in the virtual environment (right) and the corresponding paths that the user is actually steered to take in the real physical environment (left) (adapted and revised from [44]).

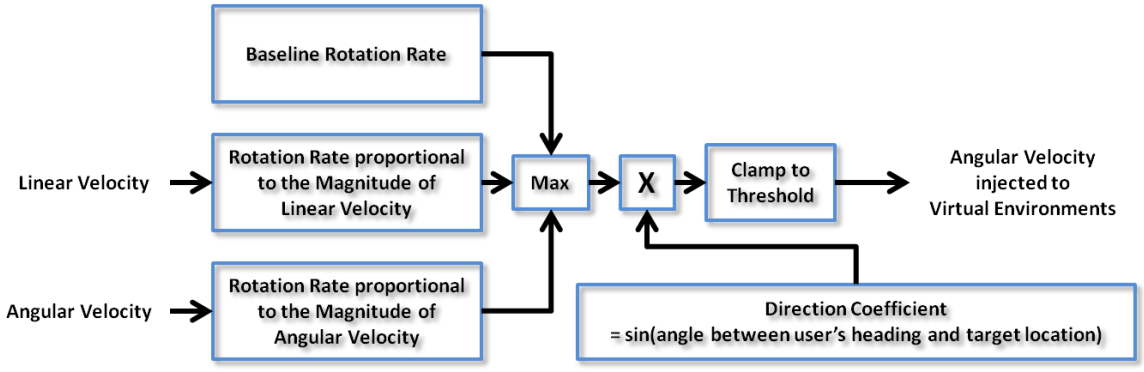


Figure 21. Block diagram of the redirected-walking mechanism (adapted and revised from [42] and [44]).

According to Figure 21, the redirected-walking mechanism uses the magnitude of user's linear and angular velocities as input parameters to determine the candidates of rotation rates to be injected into the rotation of the virtual environment. A small baseline rotation rate is injected even when the user is standing still. The maximum of the three candidates of rotation rates is chosen and scaled by a direction coefficient, which is determined by computing the sine of angle between the user's current heading and the target location in the real physical environment. Finally, the scaled value is compared with a certain threshold that is the angular velocity which seemed imperceptible to the user. If it does not exceed the threshold, it is regarded as the angular velocity to be injected into the rotation of the virtual environment. Otherwise, it is clamped to the threshold value.

According to [44], a significant issue with this mechanism is that when the user is heading directly away from the target location (often the center of the real physical environment), the system does not steer him/her back toward it. This is because the angle between the user's heading and the target location is now 180 degrees, which causes the direction coefficient equal to zero. Therefore, several additional target locations were introduced in [44] to resolve this issue. When the user is heading directly away from one target location, the system automatically selects another that is more convenient for the redirected-walking mechanism to work.

An additional mechanism proposed in [45]–[48] is to use some distractors (i.e., some things that distract) as a final effort when the user is nearly walking into walls or out of the tracking area. A distractor appears and causes the user to stop walking and to turn his/her head back and forth while watching the distractor. Meanwhile, the scene is rotated to steer the user back towards the target location.

E. SUMMARY

This chapter presented the background knowledge useful for understanding the construction of a locomotion interface for self-contained, portable, and immersive virtual environment systems.

Several of the sensor techniques were presented. The chapter discussion began with the ultrasonic sensors since the ultrasonic-based ranging measurement system was initially considered for gathering the obstacle information from the physical environment surrounding the user. Because of their greater accuracy and resolution, longer effective range, and better efficiency, the laser-based ranging sensors were then investigated as the substitutes for the ultrasonic sensors. The inertial/magnetic sensors were discussed, recognizing their uses of attitude and motion estimations that are directly related to the construction of the locomotion interface for virtual environment systems. This locomotion interface allows the user to navigate through the virtual environment by using the natural walking motions.

In order to obtain information about the physical environment for subsequent obstacle avoidance while the user is walking around, two fields of study were considered as candidates for the approaches to describe the outline of the physical environment surrounding the user and to determine the location of the user in this environment. The two fields include: (1) inertial/magnetic sensor-based personal localization, and (2) simultaneous localization and map-building.

Redirected walking was investigated for providing a mechanism to rotate the virtual environment to steer the user away from obstacles. This assumes that the required information regarding the physical environment is always obtained before calculating the injected angular velocity.

III. RANGING MEASUREMENT SYSTEM-BASED MAP-BUILDING AND LOCALIZATION ALGORITHMS

In this dissertation, the goal is to develop a locomotion interface that allows the user to use the natural walking motions to navigate through virtual environments. A means to prevent the user from walking into obstacles (such as walls) needs to be considered during the development of such a locomotion interface, since the physical environments in which the virtual environment system operates usually have limited dimensions. For implementing the obstacle avoidance task, the information constantly required may be the outline of the physical environments and the relationship between the user and such environments (i.e., the user's location in such environments) by intuition. Therefore, a ranging measurement system is adopted for its functionalities of determining the distances between the user and obstacles within a certain angular range. In this chapter, the ranging measurement system-based map-building and localization algorithms are investigated and further developed as the candidates for providing the necessary information regarding the physical environments and the user himself/herself.

A. POINT-BASED MAPS

There are various types of representations for describing a 2D map of the environment in the literature. For example, considering the shapes of different environments as well as the characteristics of different types of measurement systems, some may adopt the feature-based representation (e.g., using points, line segments, or certain shapes) for describing the outline of walls and corners in an indoor environment, while others may use the occupancy grid-based representation [49], [50] for indicating whether the space within a square is occupied or vacant in an indoor/outdoor environment.

The data collected by ranging measurement systems, such as ultrasonic or laser scanners, typically consist of a set of range measurements. Since the directions of such range measurements are known, these measurements can be represented as a set of points in the 2D or 3D coordinate system (either in polar or in Cartesian form). Therefore, considering this fact, a point-based map containing a set of points is a straightforward

representation for the range measurements. Such a map is a geometric description of the outlines of obstacles surrounding the ranging measurement system (or scanner), and it is often called a scan. An example of point-based maps (or scans) in the 2D Cartesian coordinate system is shown in Figure 22. The actual environment for this particular example is shown in Figure 23.

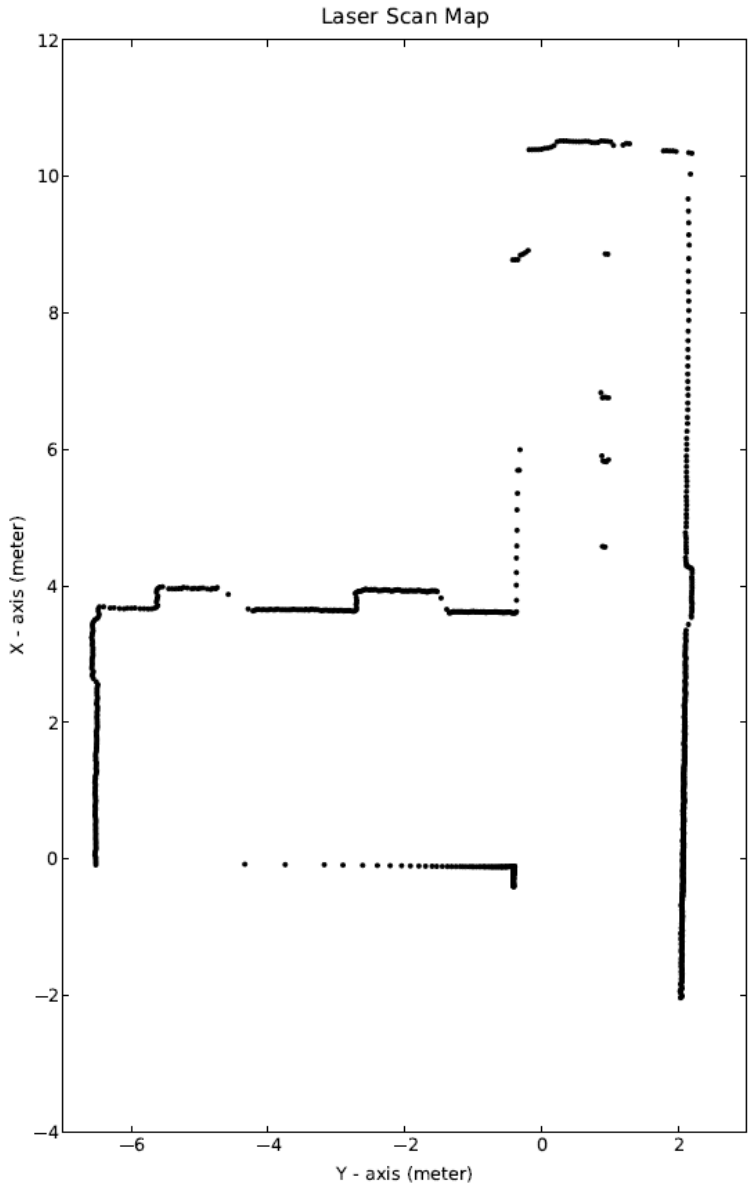


Figure 22. An example of point-based maps. The ranging measurement system is located at (0, 0) in the map consisting of a set of 1081 points. The map was taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

Figure 23. The actual environment for the example shown in Figure 22. The picture was taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

B. MATCH OF TWO POINT-BASED MAPS USING THE ITERATIVE CLOSEST POINT (ICP) ALGORITHM

As the ranging measurement system takes scans consecutively during a level movement, the map-building and localization processes can be implemented incrementally by matching and merging the older scan with every newly acquired scan over time. The scan-matching technique adopted in this research for matching two point-based maps is the iterative closest point (ICP) algorithm, which was first proposed in 1992 by Besl and McKay [51].

Supposing that there are two sets of points representing two separate scans, the scan-matching process finds a transformation that best matches one scan with the other. However, the correspondence between the two scans, which represents how the points within one scan correspond with the points within the other scan, should be found before an optimized transformation can be determined. Therefore, the correspondence and the transformation become the two main issues in the current scan-matching study [52].

The correspondence between two scans often seems fairly simple by human visual inspection based on our prior experience and knowledge about the features of

general environments where the scans may be taken. However, for an automated system, this becomes a whole new situation. Taking the point-based scans as an example, to find the correspondence between two sets of randomly positioned points is not a straightforward task. It may actually be the most difficult step in the scan-matching problem. Furthermore, in most cases, due to some amount of sensing error, the two point sets belonging to two separate scans cannot be matched exactly with each other, even if the ranging measurement system does not move while taking these scans, meaning that the true correspondence can almost never be found. To resolve this issue, some assumptions have to be made to determine a correspondence that is close enough to the truth. The ICP algorithm assumes that the correspondence between the two point sets is represented by the closest point in one set to each point in the other.

Since the assumption for determining the correspondence may not be accurate, the transformation calculated based on this correspondence assumption will not be accurate either. To resolve this, the ICP algorithm alternates between the estimation of correspondence and transformation until two scans are matched within a threshold.

Supposing that there are two scans, A and B, the ICP algorithm matching (or aligning) scan A with scan B is implemented by completing the following steps:

- For each point in scan A, assign the closest point in scan B as its corresponding point.
- Calculate the transformation (a rotation and a translation) that minimizes the sum of squared distance between every pair of corresponding points.
- Apply the transformation to scan A.
- Repeat the first three steps until scan A aligns with scan B or certain thresholds are reached.

A notional ICP process is shown in Figure 24. As illustrated in Figure 24(1) and Figure 24(3), the ICP algorithm assumes that each closest point is the corresponding point although this is often not true. Despite this, the ICP algorithm almost always brings one scan closer to the other after each iteration. More precisely, the ICP algorithm guarantees that a local minimum will be reached for the sum of squared distance between every pair of corresponding points. In some cases, the local minimum reached is also the global minimum, and the two scans are able to be aligned closely. Frequently, the process

can get trapped into one of the local minimums. More detailed information about this issue and the approaches to resolving it are discussed in the later sections.

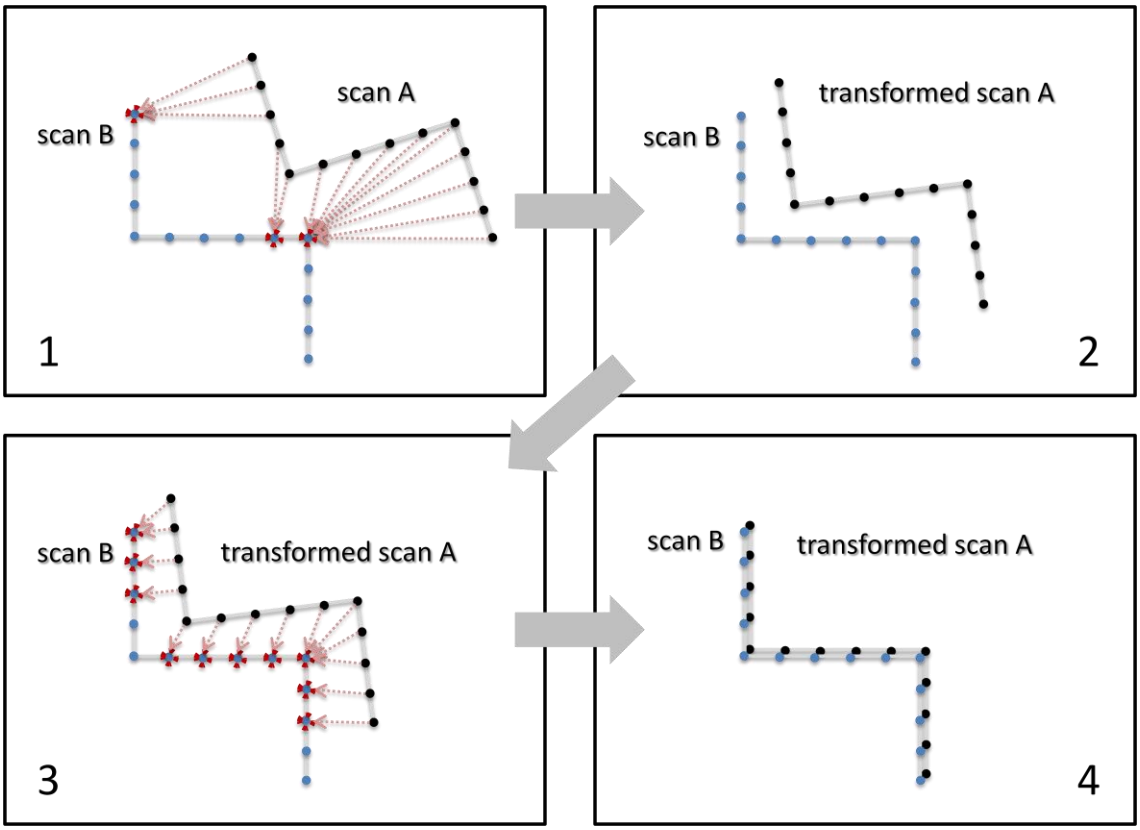


Figure 24. A notional ICP process: (1) for each point in scan A, assign the closest point in scan B as its corresponding point, (2) calculate the transformation and apply it to scan A, (3) for each point in the transformed scan A, assign the closest point in scan B as its corresponding point, and (4) calculate the transformation and apply it to the transformed scan A.

The algorithm to determine the transformation that minimizes the sum of squared distance between every pair of corresponding points was proposed in [53]. Once the correspondence between two scans is found, the optimized transformation that best matches one scan with the other can be determined by using this algorithm, which is summarized as follows [53].

Suppose that there are two corresponding point sets, A and B.

$$A = \{a_1, a_2, \cdots, a_n\}, \quad B = \{b_1, b_2, \cdots, b_n\}, \tag{3.1}$$

where a_k and b_k are 3D points, and a_k corresponds with b_k for $k = 1, 2, \cdots, n$.

$$a_k = \begin{bmatrix} a_{k,x} \\ a_{k,y} \\ a_{k,z} \end{bmatrix}, \quad b_k = \begin{bmatrix} b_{k,x} \\ b_{k,y} \\ b_{k,z} \end{bmatrix}. \tag{3.2}$$

A transformation that consists of a rotation and a translation is desired to best match A with B (or a_k with b_k). The terms $\bar{a}$ and $\bar{b}$ are the centroids of A and B, respectively.

$$\bar{a} = \frac{1}{n}\sum_{k=1}^{n} a_k, \quad \bar{b} = \frac{1}{n}\sum_{k=1}^{n} b_k. \tag{3.3}$$

The terms A' and B' are the point sets determined by referring A and B to their centroids.

$$A' = \{(a_1 - \bar{a}),\ (a_2 - \bar{a}),\ \cdots,\ (a_n - \bar{a})\} = \{a'_1,\ a'_2,\ \cdots,\ a'_n\},$$

$$B' = \{(b_1 - \bar{b}),\ (b_2 - \bar{b}),\ \cdots,\ (b_n - \bar{b})\} = \{b'_1,\ b'_2,\ \cdots,\ b'_n\}. \tag{3.4}$$

M is a 3×3 matrix whose elements are sums of products of coordinates in A' with coordinates in B'.

$$M = \sum_{k=1}^{n} a'_k {b'_k}^T = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix}, \tag{3.5}$$

where

$$S_{xx} = \sum_{k=1}^{n} a'_{k,x} b'_{k,x}, \quad S_{xy} = \sum_{k=1}^{n} a'_{k,x} b'_{k,y}, \tag{3.6}$$

and so on. N is a 4×4 real symmetric matrix whose nine independent elements represent the sums and differences of the nine elements of the 3×3 matrix M.

$$N = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix}. \tag{3.7}$$

The desired rotation represented by a unit quaternion q is to be the eigenvector corresponding to the most positive eigenvalue λ_{max} of the matrix N.

$$q = eigenvector(N, \lambda_{max}). \tag{3.8}$$

The desired translation d is the difference between the centroid of B and the rotated centroid of A.

$$d = \bar{b} - rotated\_by\_q(\bar{a}). \tag{3.9}$$

By applying the transformation (q, d) to point set A, the transformed point set A can be aligned closely with point set B.

An example of the scan-matching process using the ICP algorithm is shown in Figure 25. Ten iterations were taken for the process to reach the threshold, which was for the improvement of the latest iteration in reducing the mean squared distance between every pair of corresponding points to be less than 0.001. The result shows that the scan in blue is closely matched with the scan in red.

The test setup for the process shown in Figure 25 was designed to minimize the variability due to some outliers that will be discussed in Section C of this chapter, but not to overly idealize the test condition. The two scans were taken at exactly the same location but at different points of time to make them similar to each other but innately different (since no two scans are exactly the same due to random sensing error). Also, one of the scans (the blue one) was first rotated about its origin by 10 degrees and then translated by positive 1 meter in x-axis and positive 2 meters in y-axis. Despite how well the two scans are matched with each other, some small error may still exist in the result, and this is expected. Note that if only one scan were taken and transformed to make up the second scan, the final result of scan matching would show that the two scans ideally match with each other with no error in between because the perfect correspondence between the two scans exists in this specific case.

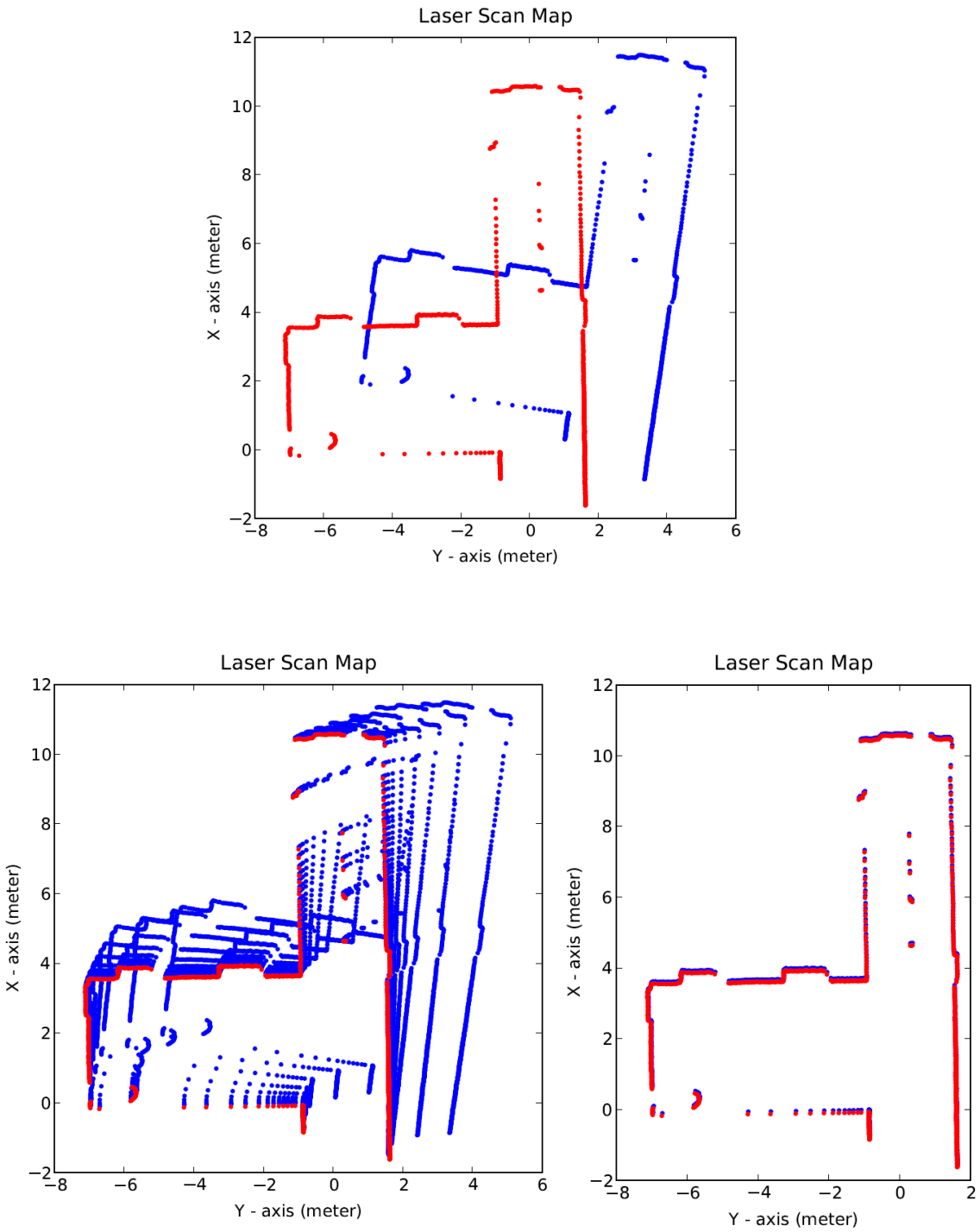


Figure 25. Scan-matching process using the ICP algorithm for two point-based scans. The upper figure shows two separate scans in blue and red, respectively. The lower-left figure shows all of the iterations for the blue scan to converge toward the red scan. The lower-right figure shows the result. Both scans consist of a set of 1081 points and were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

C. OUTLIER REJECTION FOR THE ICP ALGORITHM

For any two point-based scans to be matched, the point features that exist in one scan but have no correspondence in the other scan are treated as outliers. A notional illustration of what the outliers are referred to as is shown in Figure 26. The illustration in the figure assumes that how the two scans are supposed to be matched is known. Although this assumption is often not true, it is only intended for making the concept clearer.

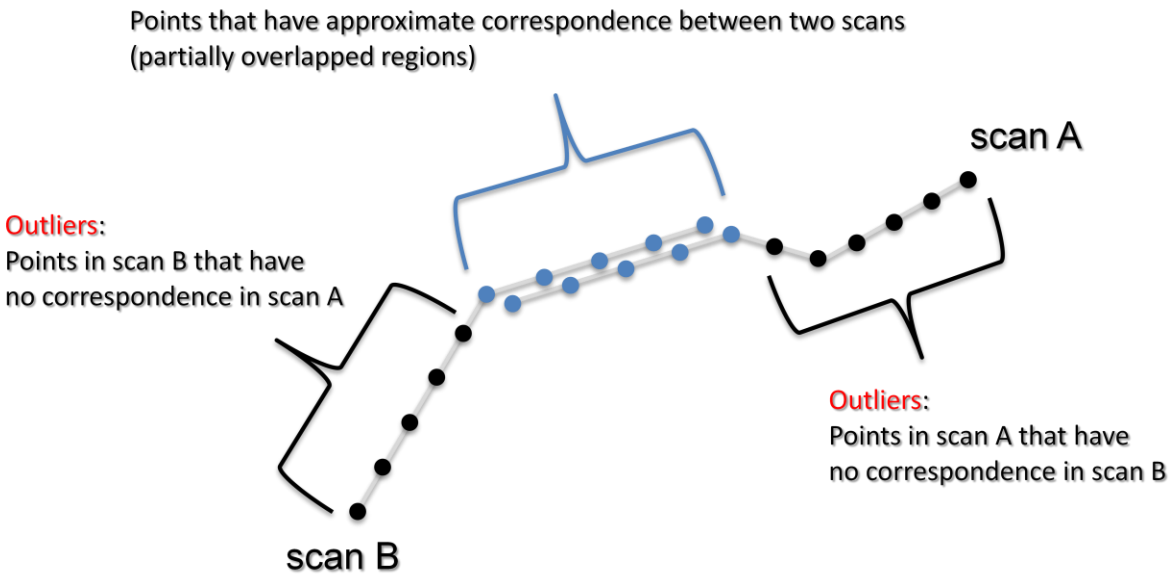


Figure 26. A notional illustration of the outliers. The points that exist in either scan but have no correspondence in the other are regarded as outliers.

Before discussing the way to identify and reject the outliers, it is important to be familiar with the effect that the outliers introduce to the scan-matching process and the reason that an outlier rejection mechanism is needed. As an example shown in Figure 27, the most apparent effect due to outliers is for the process to be trapped in a local minimum.

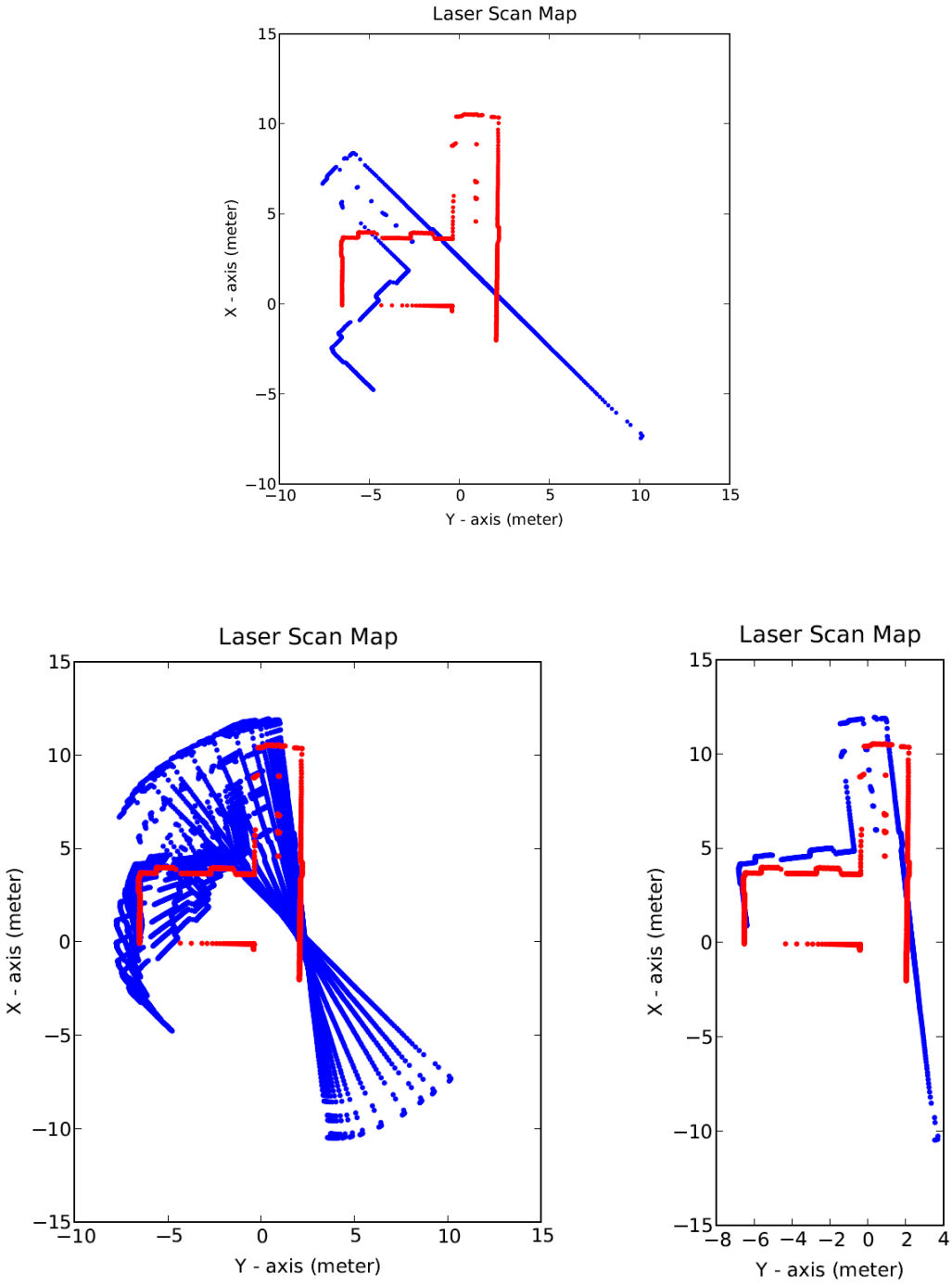


Figure 27. Scan-matching process using the ICP algorithm without rejecting the outliers. The upper figure shows two separate scans in blue and red, respectively. The blue scan consists of a set of 1047 points, while the red scan consists of a set of 1081 points. The lower-left figure shows all of the iterations for the blue scan to converge toward the red scan. The lower-right figure shows the result. Both scans were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

As mentioned earlier, the ICP algorithm determines the transformation that minimizes the sum of squared distance between each pair of corresponding points after assuming the closest-point correspondence for every iteration. While the process has found a local minimum, any transformation determined in its neighborhood during the subsequent iteration will only make the sum of squared distances larger. Therefore, the process is regarded as "trapped" in the local minimum. It is also noted that taking more iterations of the ICP process in this situation will not be helpful for one scan to converge toward the other but only toward a pose that meets the local minimum. In Figure 27, the long tail of the scan in blue is apparently a large portion of outliers causing this kind of issue, because the distance between every point on this tail and its corresponding closest point in the other scan always gets larger and thus prevents the convergence whenever the two scans tend to converge. This suggests that the local-minimum issue can be resolved to a certain degree by rejecting some of the outliers.

There are various techniques in the literature that deal with the outlier issue. For instance, Phillips, Liu, and Tomasi [54] developed a distance measure called fractional root mean squared distance (FRMSD). The transformation determined by minimize this distance value between two scans was reported to be less susceptible to outliers. This was under the assumption that the points from two separate scans are similar and have a strong correspondence with each other. In other words, the transformation between two scans needs to be relatively small, and so is the outlier portion. As the number of outliers increases, some rejection mechanism is still needed to ensure a successful scan-matching process.

Rusinkiewicz and Levoy [55] approached the outlier rejection study through justifying the distance between each pair of corresponding points. Under a similar assumption that the transformation between two scans is relatively small, each pair of corresponding points are within a certain distance from each other. The points with their correspondence found outside a user-specified distance are therefore regarded as outliers and are rejected. This technique was adopted and tested in the ICP algorithm when our research initially approached the outlier rejection issue. However, it does not provide a marked effect, and the process may still be trapped in the local minimum as shown in the

example in Figure 27. A more tangible technique is therefore needed to be incorporated in the ICP algorithm to better cope with this issue.

In [56] and [57], the frustum culling approach was adopted. A frustum represents the field of view and is defined through this approach. The points that fall outside the frustum are regarded as outliers and are neglected.

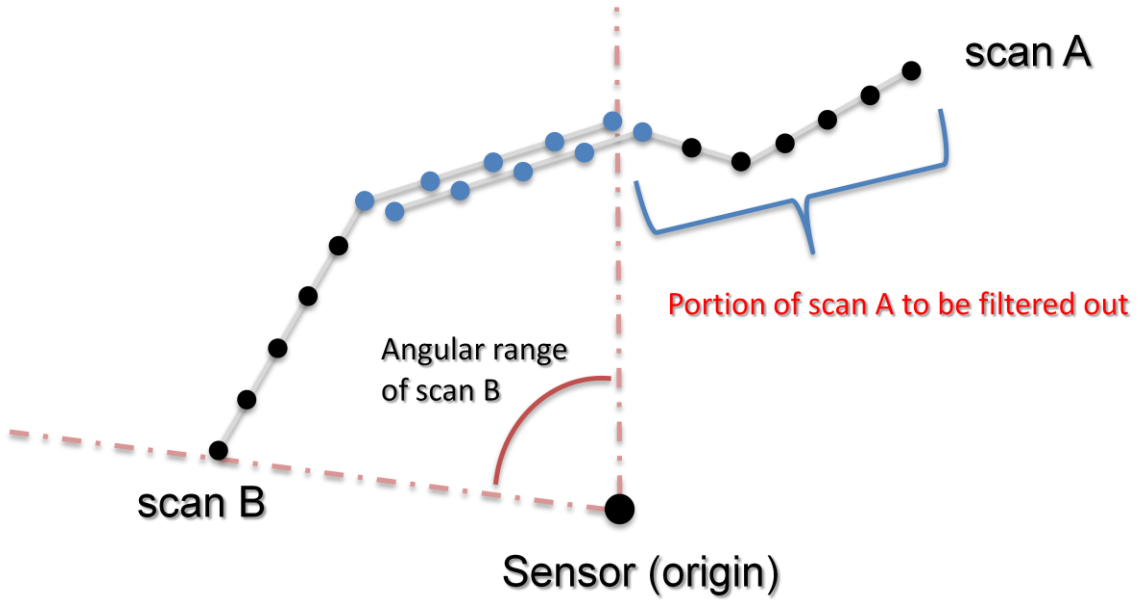


Figure 28. A notional illustration of the portion of outliers to be filtered out. The portion of scan A that falls outside the angular range of scan B is to be filtered out. In the figure, the angular range is referred to as the angular viewing range of scan B from the sensor viewpoint where scan B was taken, and the sensor (origin) point represents the position of the ranging measurement system.

Similar to the frustum culling approach, the outlier rejection mechanism proposed in this research utilizes the angular range of the scan to be matched with. The angular range mentioned here is referred to as the angular viewing range of the scan from the viewpoint of the ranging measurement system where the scan was taken. Assuming that the scan-matching process is to match scan A with scan B as shown in Figure 28, the portion of scan A that falls outside the angular range of scan B is regarded as the outlier portion and is filtered out before assigning the correspondence for each ICP iteration. Note that in order to better explain the concept of the filtering mechanism, the illustration

in the figure shows a final scan-matching stage at which the two scans are almost aligned together and most of the outliers can be identified and filtered out. On the other hand, at the beginning of the scan-matching process, because the two scans are not yet matched, there may only be a small portion or none of the outliers being identified. As the process continues, the outlier portion being identified will increase incrementally. Also, it is important to point out that the outlier rejection mechanism is only implemented for finding the correspondence between the two scans and calculating the optimized transformation for each ICP iteration. The transformation determined is then applied to all the points in the original scan. There will be no point actually being removed from the scan.

Figure 29 shows the scan-matching process using the ICP algorithm combined with the outlier rejection mechanism proposed in this section for the two scans used in Figure 27. A total of 19 iterations were taken for the process to reach the threshold, which was for the improvement of the latest iteration in reducing the mean squared distance between every pair of corresponding points to be less than 0.001. The issue that the outliers caused for the scan-matching process to be trapped in a local minimum can be resolved at least to a certain degree (if not for all the cases) by combining this outlier rejection mechanism. The two scans in the figure are able to be aligned closely with each other.

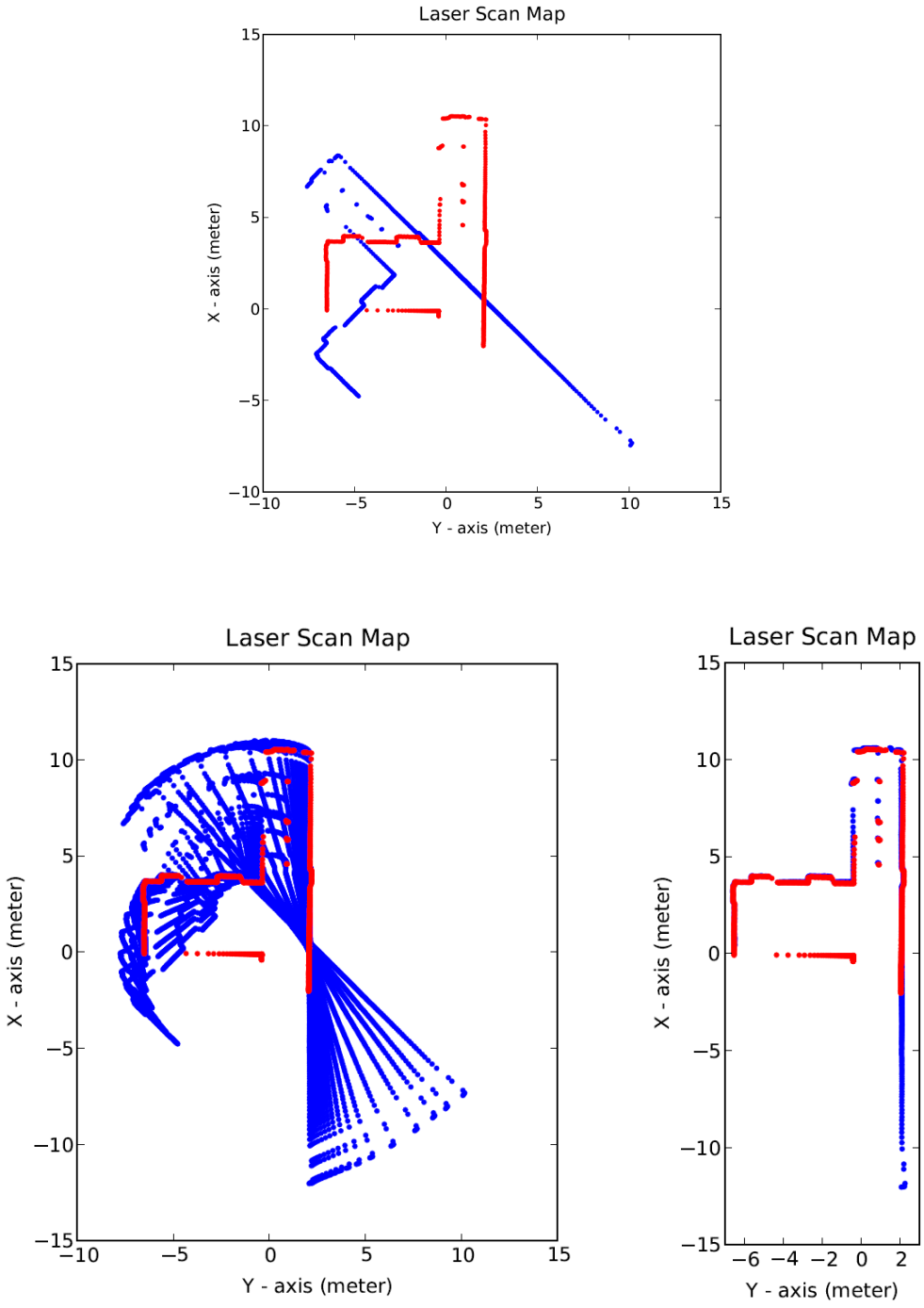


Figure 29. Scan-matching process using the ICP algorithm combined with the outlier rejection mechanism proposed in this research. The upper figure shows two separate scans in blue and red, respectively. The blue scan consists of a set of 1047 points, while the red scan consists of a set of 1081 points. The lower-left figure shows all of the iterations for the blue scan to converge toward the red scan. The lower-right figure shows the final result. Both scans were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

D. MERGENCE OF TWO MATCHED POINT-BASED MAPS

The map-building and localization algorithm implemented in this research features a scan-matching process that matches an older scan with a newly acquired scan. The older scan mentioned here can represent either a single scan that was taken earlier or a composition of a series of scans taken so far. For an incremental process that builds the map and determines the whereabouts of the ranging measurement system that takes scans, the latter definition is referred to. However, the same process applies to either one.

Therefore, after every completion of the scan-matching process, some means is needed to merge the two matched scans into a single one. The merging method depends on the requirements we set for the process, such as the type of implementation (real time or post processing), and the intended use of the map being built.

Since there is a large overlapped portion between the two matched scans in a typical continuous scanning process, keeping all of the points when the two scans are merged into a single one is not efficient. The growth of the number of redundant points will considerably reduce the efficiency of the subsequent scan-matching process and prevent it from being implemented in real time. To resolve this issue, a method featuring the idea of sparse point maps was proposed in [57]. It is intended to avoid duplicate storage of points by conducting an additional correspondence search and neglecting the points from one scan that correspond to the same points already existing in the other scan. A drawback of this method is that the additional correspondence search is time consuming since it needs to be applied to all points, not a down-sampled subset of points as often used during each ICP iteration.[1]

For this research, the merging method consists of several simple rules of how the two scans are to be merged and is described as follows. A notional illustration of the merging process following these rules is shown in Figure 30.

- Always keep the new scan points. For the application in this research, the map to be built is mainly used for obstacle avoidance. Therefore, the accuracy for the portion of the map around the ranging measurement system carried by the user is more important than that for the portion far

[1] Down-sampling the scan points for correspondence search can effectively reduce the time consumption of each ICP iteration. This will be discussed in Section E of this chapter.

away. Since the newly acquired scan is always more accurate as opposed to the older scan containing a larger accumulated error and thus represents the environment around the user better, the merging process always keeps the new scan points.

- Discard the points from the transformed old scan that lie within the overlapped portion of two scans (or within the angular range of the new scan). For the real-time implementation in this research, the process is allowed to discard a portion of old information to avoid unlimited growth of the data amount that slows down the whole process over time. Therefore, partly following the first rule, the points from the transformed old scan that lie within the angular range of the new scan become redundant and need to be removed.
- Keep the points from the transformed old scan that lie outside the overlapped portion of two scans (or outside the angular range of the new scan). These are the only points from the transformed old scan being kept since the others were discarded as described in the previous rule.

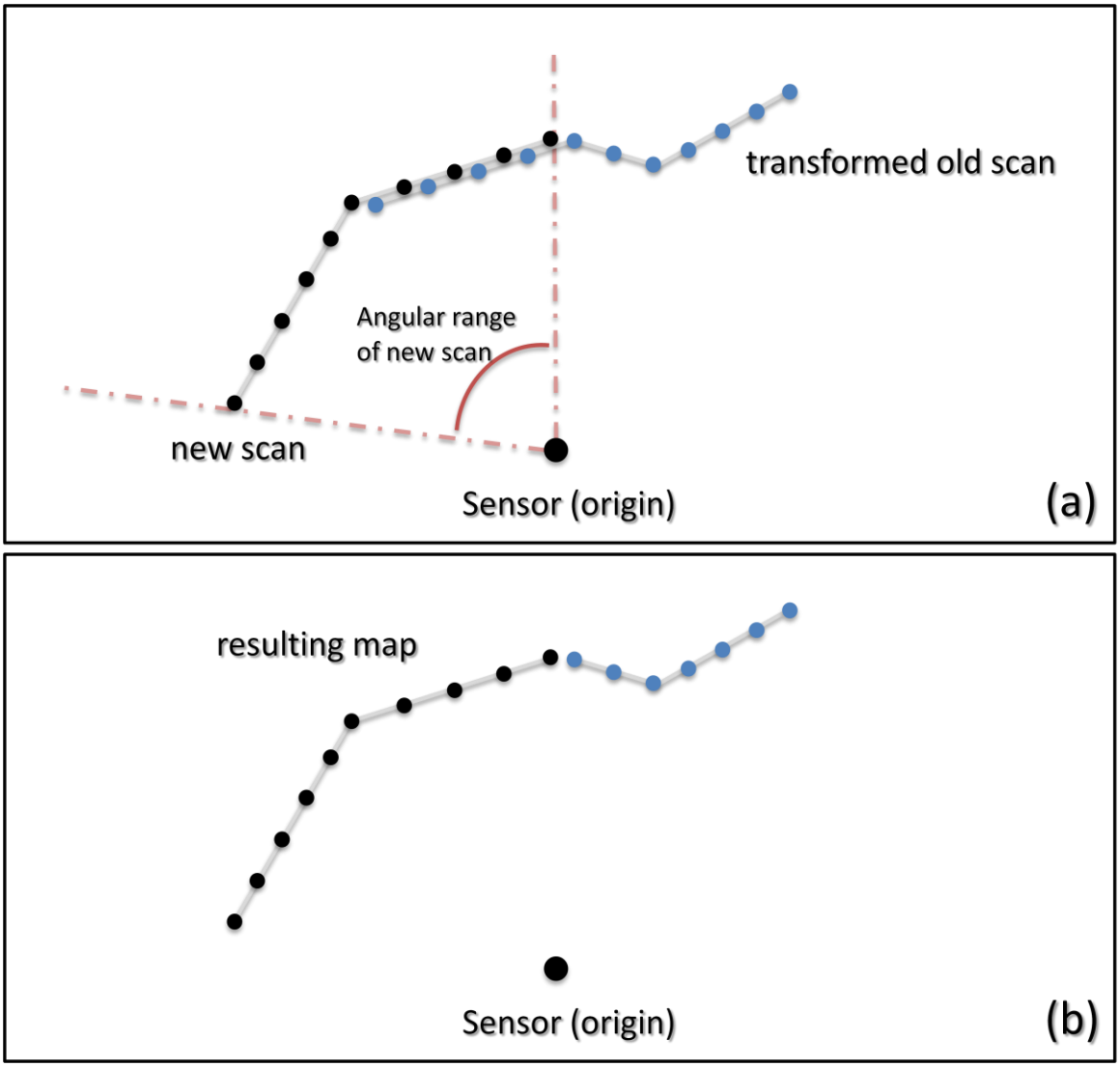


Figure 30. A notional illustration of the merging process. Two matched point-based scans being merged are shown in (a). Following the merging rules, the resulting map (scan) is shown in (b). In the figure, the angular range is referred to as the angular viewing range of the new scan from the sensor viewpoint where the new scan was taken, and the sensor (origin) point represents the position of the ranging measurement system.

An example of merging two matched scans following the three simple rules is shown in Figure 31. In the figure, the scan in red represents the transformed old scan, while the scan in blue represents the newly acquired scan. The resulting scan in black is obtained by merging the two matched scans.

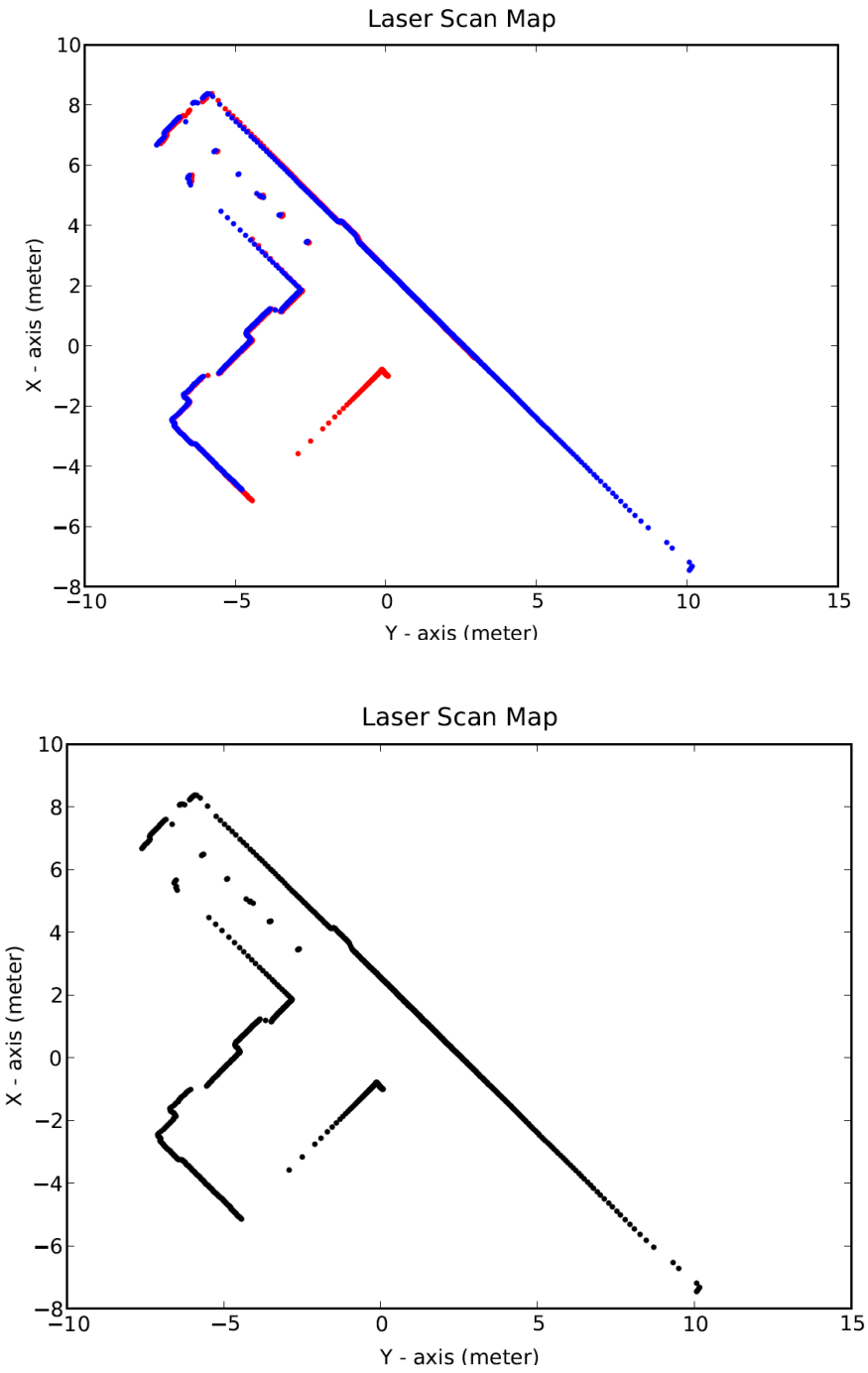


Figure 31. An example of merging two matched scans following the three simple rules. The upper figure shows two matched scans. The scan in red represents the transformed old scan consisting of 1081 points, while the scan in blue represents the newly acquired scan consisting of 1047 points. The lower figure shows the resulting scan obtained by merging the two matched scans. Both scans in the upper figure were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School. The resulting scan consists of a set of 1242 points.

E. REAL-TIME IMPLEMENTATION OF THE POINT-BASED MAP-BUILDING AND LOCALIZATION PROCESS

Based on the presentation in the previous sections, it is noted that the two separate processes in this research, map-building and localization, are closely coupled with each other, and can be combined and treated as one single process. The incremental process for point-based map-building and localization in this research usually starts with the first scan taken by the ranging measurement system. The first scan is regarded as the original old scan. (It should be noted that a known map built beforehand can also be adopted as the original old scan, as long as it has the same representation as the one used in this process. In such a point-based process, it needs to be a point-based map.) For every new scan acquired, the process cycle completes the following two steps:

- Match the old scan with the newly acquired scan using the ICP algorithm.
- Merge the two matched scans into one single scan, which in turn becomes the newer old scan.

As the process matches the current old scan with the newly acquired scan, the transformation for the old scan to be matched with the new scan is determined at the same time. Such transformation at the end of the kth cycle consists of a rotation q_k (represented by a unit quaternion) and a translation d_k (represented by a 3D vector). The orientation Q_k and position D_k of the ranging measurement system after the completion of the current (kth) process cycle relative to its initial orientation Q_0 and position D_0 can be determined as follows.

$$Q_k = q_k^* Q_{k-1},$$
$$D_k = Q_k(-d_k)Q_k^* + D_{k-1}. \tag{3.10}$$

The orientation Q_k and position D_k are represented by a unit quaternion and a 3D vector, respectively. The subscript k represents that it is the kth process cycle, and the superscript * represents the complex conjugate of the given quaternion. The operation shown in the equations is the quaternion multiplication. Note that the following values are adopted for the initial orientation Q_0 and position D_0.

$$Q_0 = [1.0, 0.0, 0.0, 0.0],$$
$$D_0 = (0.0, 0.0, 0.0). \tag{3.11}$$

The absolute initial orientation and position values may be applied instead. In this case, the terms Q_k and D_k will be representing the absolute values as well.

After the two matched scans are merged, the resulting scan obtained at the end of current cycle is represented in a coordinate relative to the current orientation and position of the ranging measurement system (i.e., represented in the ranging measurement system's coordinate). However, since the orientation Q_k and position D_k of the ranging measurement system are known, the coordinate system of the scan can be converted into the one relative to the initial orientation and position of the ranging measurement system, if needed. This can be completed simply by applying a rotation (with the same value as the orientation Q_k) and a translation (with the same value as the position D_k) to all of the points in the resulting scan.

A general concept of real-time implementation is often referred to as an algorithm that is able to be implemented to process the raw sensor data as they are gathered without the need of any buffering mechanism. The process introduced in this chapter takes a new scan of the environment and conducts scan matching (and merging) between the old and new scans for each process cycle. It is considered real-time if it is able to be implemented continuously while the ranging measurement system is moved. Since the moving speed of the system is as low as humans' normal walking speed, it is thus reasonable for the real-time process to allow one to two seconds on average for each of the process cycles. Note that the time taken for the process to take a new scan is in tens of milliseconds, which is negligible as compared with that to complete scan matching and merging.

For the real-time implementation of the point-based map-building and localization process, the process efficiency is the main issue that needs to be addressed. In other words, the time consumption for each process cycle needs to be reduced so that it operates reasonably fast for the real-time implementation. It is noted that the most time-consuming portion in current process cycle is the portion in the ICP algorithm that searches for the correspondence between the two scans being matched.

When a brute-force method is used, the way of searching for the closest-point correspondence will be to calculate and compare all of the squared distances from all of the points in the newly acquired scan to each point in the old scan. In this way, if the

numbers of points in the old scan and the newly acquired scan are M and N respectively, the complexity of this search for each ICP iteration of the process cycle will be $O(MN)$, which is fairly computationally expensive for scans with large sizes.

In dealing with the complexity issue, a method utilizing a tree-search technique (called K-D tree) was proposed by Maneewongvatana and Mount [58] in 1999. By incorporating this method to the ICP algorithm, searching for the closest points among a set of points can be implemented more efficiently. The general idea of the K-D tree is to form the data points into a binary tree, each of whose nodes specifies an axis and splits the points based on their coordinates along that axis. The method constructs a K-D tree structure for the points of the newly acquired scan. During each ICP iteration, the process searches for the closest point within this K-D tree to each point in the old scan. The complexity for each closest-point search is $O(log_2 N)$ if there exist N points in the newly acquired scan. For the old scan containing M points, the complexity for the complete correspondence search is, therefore, $O(Mlog_2 N)$, which turns out to be less computationally expensive than the brute-force method. Note that the complexity of constructing the K-D tree is $O(Nlog_2 N)$. It only needs to be completed once during each process cycle.

Despite the improvement in efficiency obtained through the use of the K-D tree search method, the time consumption of each process cycle is often still too high for the process to be implemented in real time, especially when the sizes of the two scans to be matched are large. Therefore, in addition to the K-D tree-search method, some other means is still needed to reduce the computational load of the correspondence search. The approach adopted in this research is to down-sample the old scan by some number at the beginning of every ICP iteration to reduce the number of points in the old scan participating the correspondence search. This resolves the efficiency issue to a certain degree. However, down-sampling the old scan may also reduce the scan-matching accuracy if the remaining point features cannot offer enough information for the process. This may also increase the chance for the scan-matching process to be trapped in a local minimum and thus extra care should be taken. As mentioned earlier, there is always an

accuracy-efficiency trade-off for almost all of the existing algorithms. This is one of the situations in which a trade-off should be determined.

An example for one of the map-building and localization process cycles is shown in Figure 32. In this particular case, a comparison is made to show the time consumption of the map-building and localization process with different correspondence search methods and setups incorporated into the ICP algorithm. The comparison is shown in Table 1. It is noted that the time consumption for the process can be reduced so that it can be implemented in real time by down-sampling the old scan and incorporating the K-D tree correspondence search method into the ICP algorithm. Note that while the threshold value is specified as the improvement of the latest iteration in reducing the mean squared distance between every pair of corresponding points, adopting a larger threshold value can reduce the time consumption as well. However, this will directly degrade the scan-matching accuracy and will not be emphasized in this research.

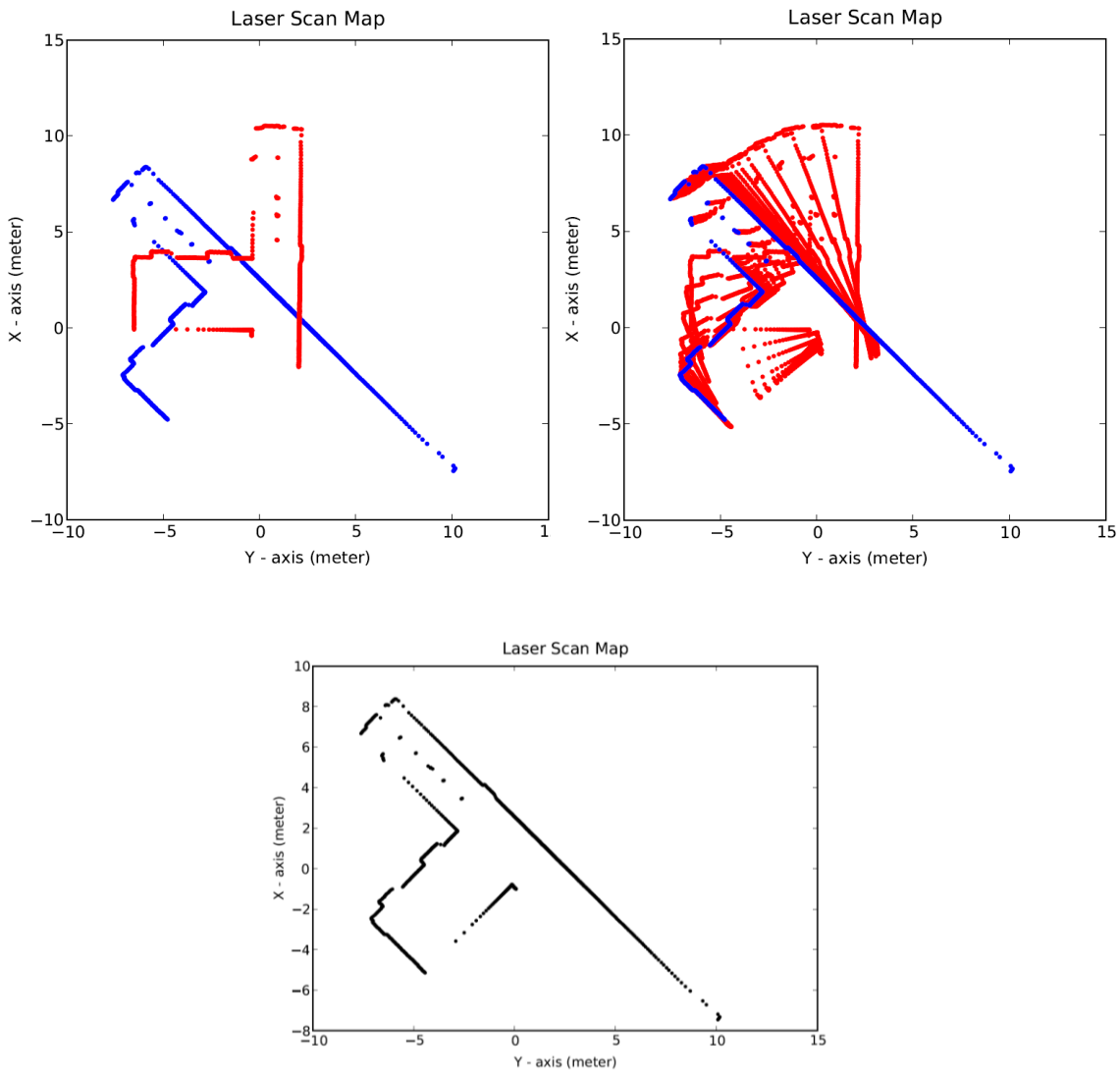


Figure 32. An example of the map-building and localization process cycle. The upper-left figure shows two separate scans. The one in red is the old scan consisting of 1081 points, and the one in blue is the new scan consisting of 1047 points. The upper-right figure shows all of the iterations for the old scan to converge toward the new scan. The lower figure shows the resulting scan obtained by merging the two matched scans. Both scans in the upper-left figure were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School. The resulting scan consists of a set of 1242 points.

Table 1. A comparison in time consumption of map-building and localization process with different correspondence search methods and setups incorporated into the ICP algorithm (for a particular case the same as in Figure 32). The program was implemented by using the Python script language and executed on a laptop with an Intel Core i7 2.20GHz processor. The code in the program is not optimized, and the value produced is only intended for an approximate comparison.

Methods Incorporated	# of Iterations	Time (Seconds)
Brute-force method	19	127.49573576200
K-D tree method	19	10.70492927460
Down-sampling the old scan by 20 + K-D tree method	20	1.13340847026

Threshold (Meters^2):
Improvement of the latest iteration in reducing the mean squared distance between every pair of corresponding points <= 0.000025

An example of the real-time point-based map-building and localization process introduced up to this point is shown in Figure 33. The ranging measurement system was moved approximately in a rectangular loop (i.e., moved alternately between moving forward and turning left until back to the start position). The point set in blue represents the map of the environment, while the point set in red shows the trajectory of the ranging measurement system, which indicates the locations where the scans were taken. The map was built by consecutively matching and merging the scans taken by the ranging measurement system as it was moved. The coordinates of the map are relative to the initial position and orientation of ranging measurement system that follow the north-east-down (NED) coordinate system. In our specific case representing the 2D map, a coordinate system consisting of only the north and east axes (x and y) is used instead. A picture of the actual environment taken from the lower-right corner in the map is shown in Figure 34.

As being observed in Figure 33, it is difficult to tell the estimation error for the transformation from the final map. However, it can be reflected by the estimated trajectory of the ranging measurement system. Although the actual initial and final positions were exactly the same, the estimation result still shows that there is a large error between the initial and the final positions. This is due to the errors accumulated during

the estimation of transformation in each of the scan-matching processes. Some other means is still needed to effectively deal with this issue in a sense that either the innately unbound error can be greatly reduced or it can be bound.

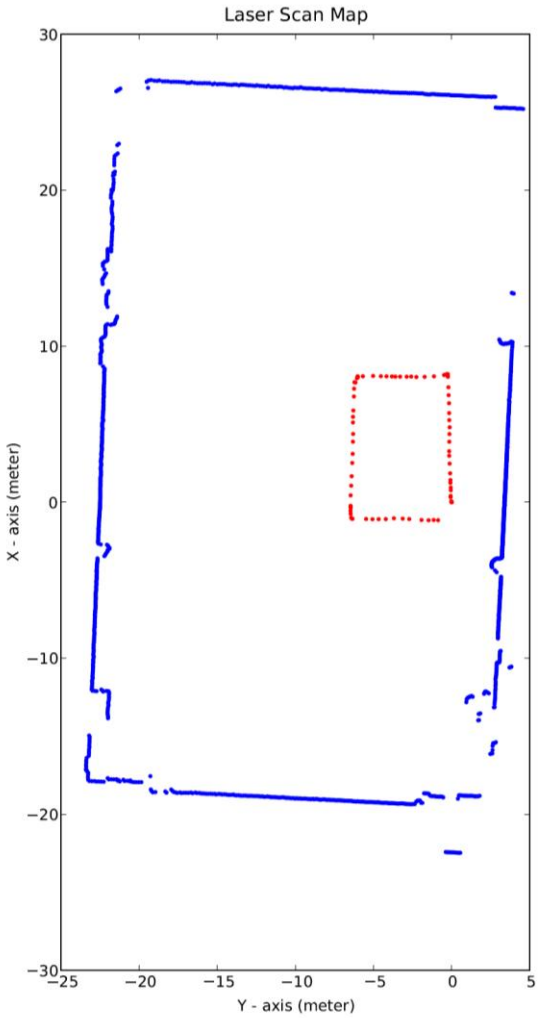


Figure 33. An example of the real-time point-based map-building and localization process. The point set in blue represents the resulting map of the environment, while the point set in red shows the trajectory of the ranging measurement system. The map consisting of 1473 points was built by matching and merging 81 individual scans. The coordinate system of the map is relative to the start position and orientation of ranging measurement system. Although the actual initial and final positions were both (0, 0), the estimation result shows that the initial position (0, 0) and the final position (-1.15098078202, -0.874567350456) do not match due to the errors accumulated during each of the scan-matching processes. The test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

Figure 34. The actual environment for the process shown in Figure 33. It is an indoor basketball court of Miami University in Oxford, Ohio.

F. LINE-BASED REPRESENTATION FOR THE ORIGINAL POINT-BASED MAPS

1. Motivation

The point-based representation is able to sufficiently describe the outline of the environment where the ranging measurement system operates as shown in several of the previous examples. However, as a means of dealing with the issue of a large accumulated estimation error, some other type of representation is also worth considering in a sense that it may provide better information than the original point-based map for correcting the error to a certain degree.

The line-based representation that uses line segments to describe the outlines of environments is adopted in this research, based on the following reasons:

- The outlines of most indoor environments from the top view can often be described by a set of 2D line segments.
- Line segments are better able to represent boundaries (e.g., walls) than points. For point-based representation, it is unknown whether the space between any two adjacent points is vacant or not.

- The use of line segments is computationally more efficient than points, since each line segment is able to represent all those points through which it goes.
- When implementing the scan-matching process using the ICP algorithm, instead of including all of the points from the original point-based scan, adopting the start and end points of each line segment from the converted line-based scan is a means to a better efficiency, since this indirectly but systematically down-samples the original scan for each ICP iteration.
- Built on top of the ICP algorithm, some mechanism utilizing the line features is expected to correct the estimation error for the transformation in each of the scan-matching process.

2. Hough Transform

One of the basic techniques that utilize the line features is the Hough transform [59]. The basic idea is summarized as follows.

As shown in Figure 35 (a), a straight line in the 2D coordinate system can be described by the following equation.

$$y = mx + b. \tag{3.12}$$

where m is the slope parameter, and b is the intercept parameter.

The line is thus represented by the parameters (m, b). One issue of this representation is that it is not stable. When the line gets horizontal, the values of m and b become infinite. To resolve this, a more useful representation is used.

As shown in Figure 35 (b), the same straight line can also be described by using the range r and the bearing θ. The range r is the distance between the origin and the line, while the bearing θ is the angle of the vector from the origin to its closest point on the line.

$$r = xcos\theta + ysin\theta. \tag{3.13}$$

Assuming that two points (x_1, y_1) and (x_2, y_2) on the straight line are known, the bearing θ and the range r can be calculated as follows [60].

$$\theta = \begin{cases} \tan^{-1}\left({(x_2 - x_1)}/{(y_1 - y_2)} \right), & if \;\; y_1 \neq y_2. \\ {\pi}/{2}\,, & if \;\; y_1 = y_2. \end{cases} \tag{3.14}$$

$$r = x_1 \cos\theta + y_1 \sin\theta = x_2 \cos\theta + y_2 \sin\theta. \tag{3.15}$$

Therefore, the line can be represented by a unique point in the r-θ parameter space (i.e., in the Hough transform) as shown in Figure 35 (c). A useful concept of using this representation for lines is that when two or more points in the r-θ parameter space are close enough to one another, they can be described approximately by the same straight line in the x-y space.

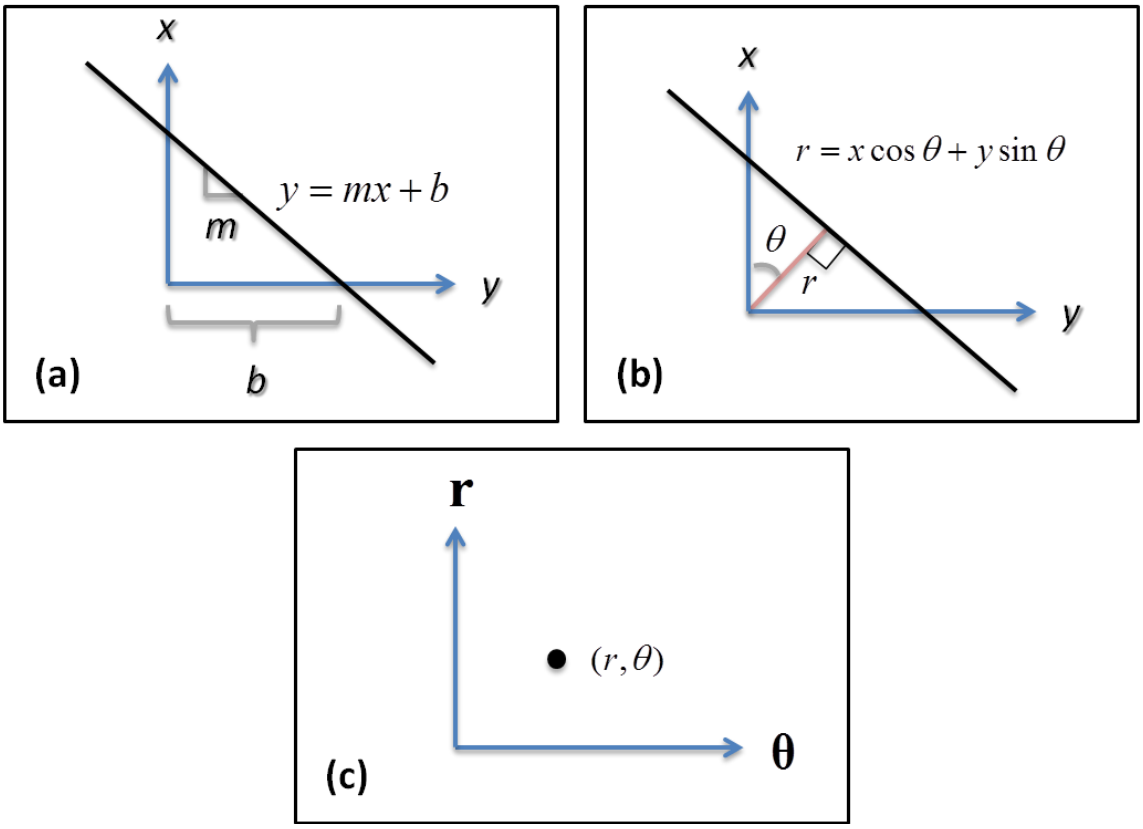


Figure 35. A straight line represented in different coordinate systems: (a) Cartesian coordinate system, (b) Cartesian coordinate system, and (c) r-θ parameter space (polar coordinate system).

3. Conversion of Point-Based Maps into Line-Based Maps

As mentioned earlier, the data collected by the ranging measurement system consist of a set of range measurements. Since the directions of such range measurements are known, these measurements can be represented as a set of points in the 2D or 3D coordinate system. A point-based map containing a set of points is therefore a fairly straightforward representation for such range measurements. However, a line-based representation is considered to further deal with the issue that may not be directly handled as a point-based map is used. It is thus necessary to convert the original point-based map into a line-based map.

Given a point-based map that consists of a set of points, the steps for building a line-based map are as follows. A notional illustration is shown in Figure 36.

- Connect any two consecutive points by a line segment if they are close enough to each other to represent a portion of the boundary. A distance threshold can be specified to decide whether the space between any two consecutive points should be left vacant. Each of the line segments has the following attributes: the start and end points, the bearing θ, the range r, and the line length.
- Examine the r-θ parameter space of any two adjacent line segments to determine whether the points they represent are close enough to be combined, and keep track of these line segments. A distance threshold can also be specified. Clusters of points that are closer to one another than the threshold value in the r-θ parameter space represent sets of line segments to be combined and approximated by a common line segment for each set.
- For every set of line segments to be approximated by a common line segment, extract the original points corresponding to the line segments and compute the least square line segment that fits these points. All of the resulting least square line segments will replace the original line segments for representing the line-based map.

An example of converting a point-based map into a line-based map is shown in Figure 37. The upper figure shows the original point-based map while the lower figure shows the converted line-based map. The lengths and the number of line segments in the resulting line-based map depend on the threshold we set for the original short line segments to be combined and approximated by a common line segment. The threshold value specified in this research is the upper bound of the distance between any two points (1.0 for this particular example) in the r-θ parameter space being considered short

enough for the two points to be regarded as a single one. If any two points are closer to each other than this distance, they represent the same line in the x-y space. It is noted that representing the same map by using line segments can greatly reduce the space needed for storing and maintaining the map, since each line segment can replace a certain number of the original points.

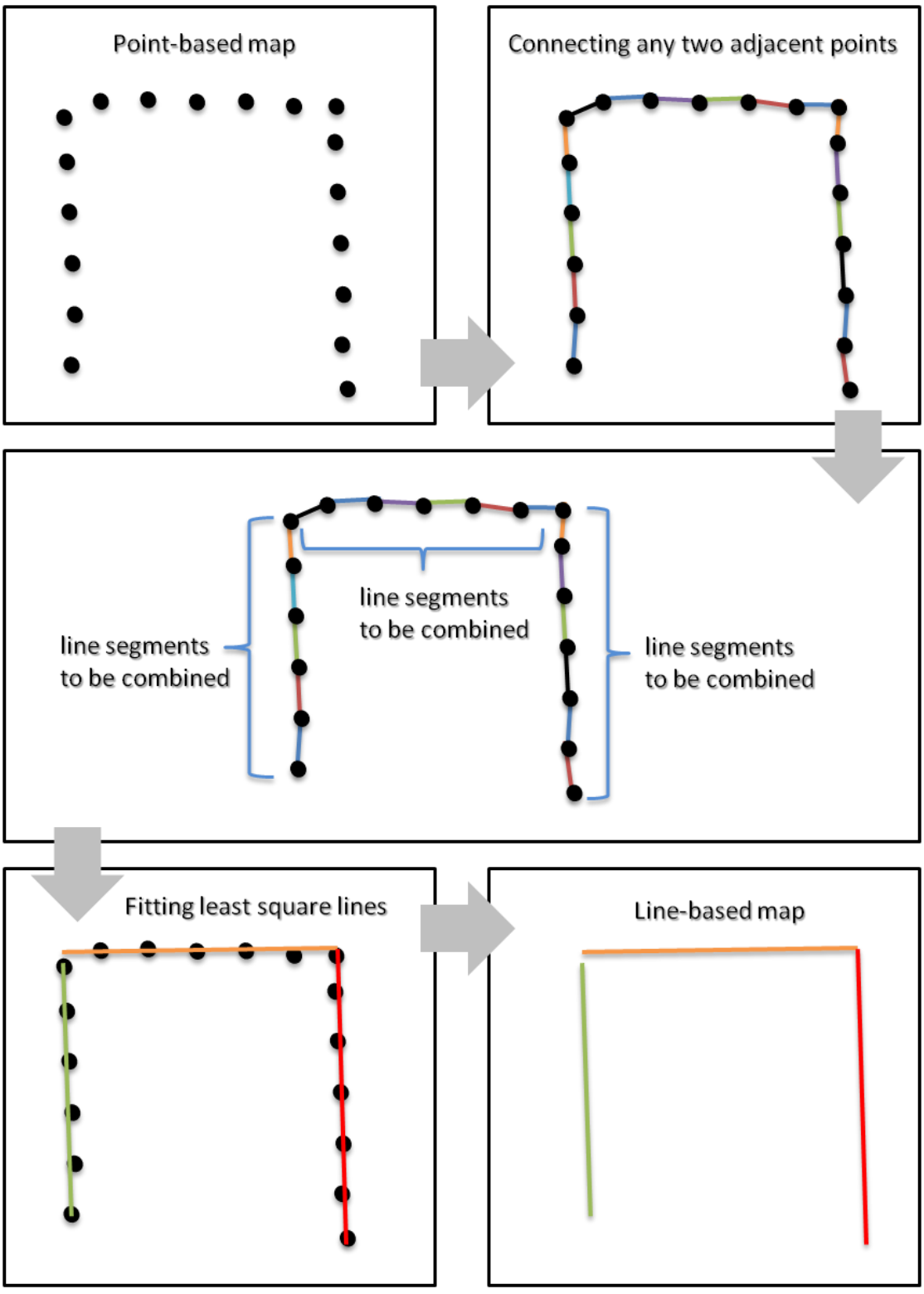


Figure 36. A notional example for steps of converting a point-based map into a line-based map.

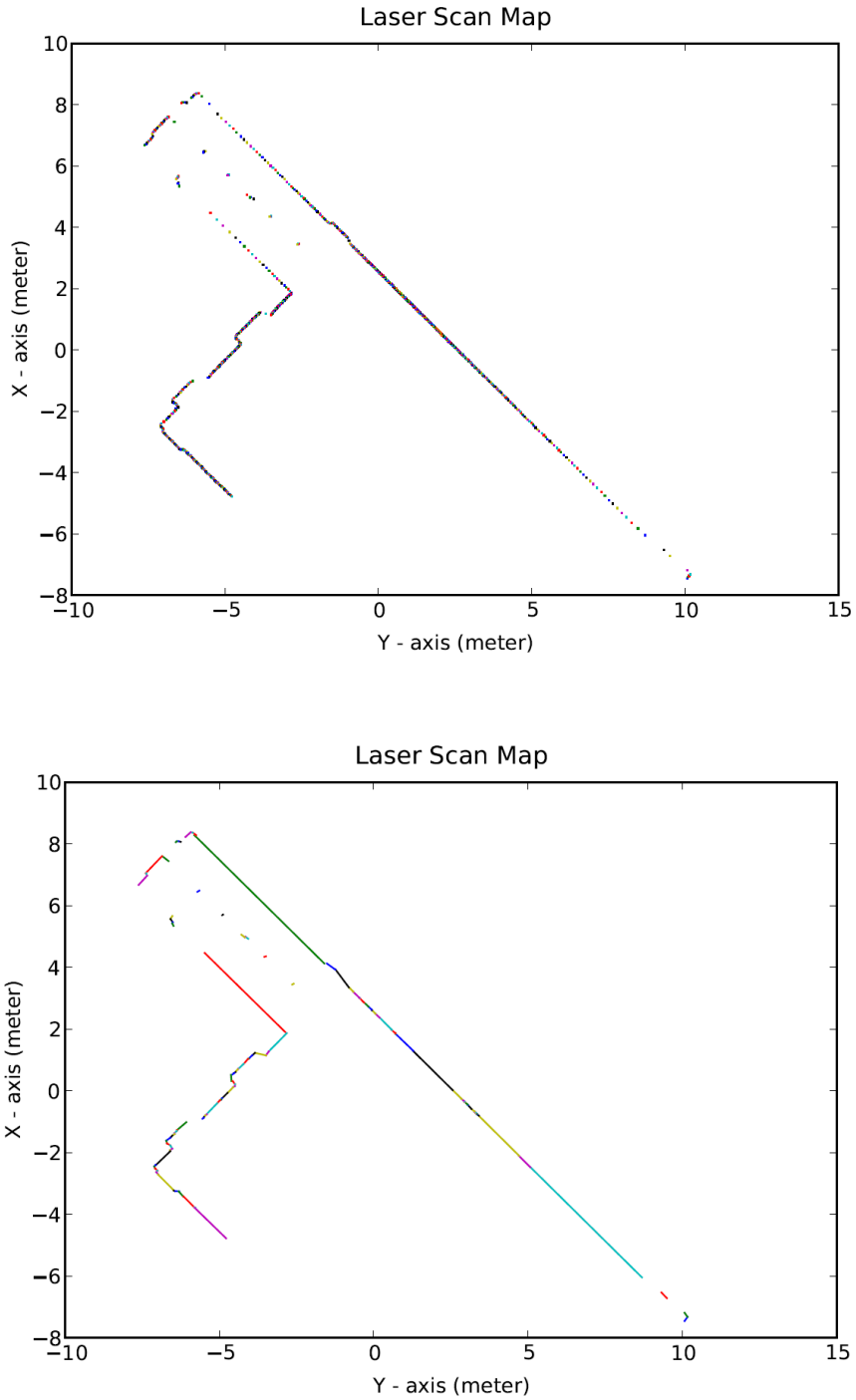


Figure 37. An example of converting a point-based map into a line-based map. The upper figure shows the original point-based map consisting of 1047 points. The lower figure shows the converted line-based map consisting of 175 line segments. The scan was taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

G. REAL-TIME IMPLEMENTATION OF THE LINE-BASED MAP-BUILDING AND LOCALIZATION PROCESS

1. Match of a Line-Based Map with a Point-Based Map using the ICP Algorithm

The process for the line-based map-building and localization is to build a line-based map incrementally and to estimate the transformation at the same time during the implementation of scan matching. Since every newly acquired scan is represented by a set of points (i.e., a point-based map), it is thus necessary to match the old line-based map with the newly acquired point-based map.

Since the line-based map containing a set of line segments can be easily converted into a point set by extracting the start and end points from each of the line segments, the ICP algorithm used for matching two point sets can therefore be adopted for the scan-matching process between a line-based map and a point-based map. Because of the significantly reduced number of points in the line-based map, the efficiency for the ICP process is expected to be better than originally matching two point-based maps. The steps for matching a line-based map with a point-based map are as follows.

- Extract the start and end points from each of the line segments in the line-based map to form a point set.
- Match the point set with the newly acquired point-based map using the ICP algorithm.
- Apply the transformation calculated during matching the point set with the point-based map to the line-based map.

An example of matching a line-based map with a point-based map is shown in Figure 38. The map in blue represents the point-based map while the map in red represents the line-based map. Using the same raw sensor measurements as in this particular example, a comparison is made to show the time consumption while implementing two different types of scan-matching processes. One is to match two point-based maps, while the other is to match a line-based map with a point-based map. Both cases are implemented without down-sampling the maps for each correspondence search. The comparison is shown in Table 2. It is noted that matching a line-based map with a point-based map is more efficient than matching two point-based maps. For real-time implementation, the time consumption for both cases can be further reduced by applying

appropriate down-sampling to the maps for the correspondence search of each ICP iteration.

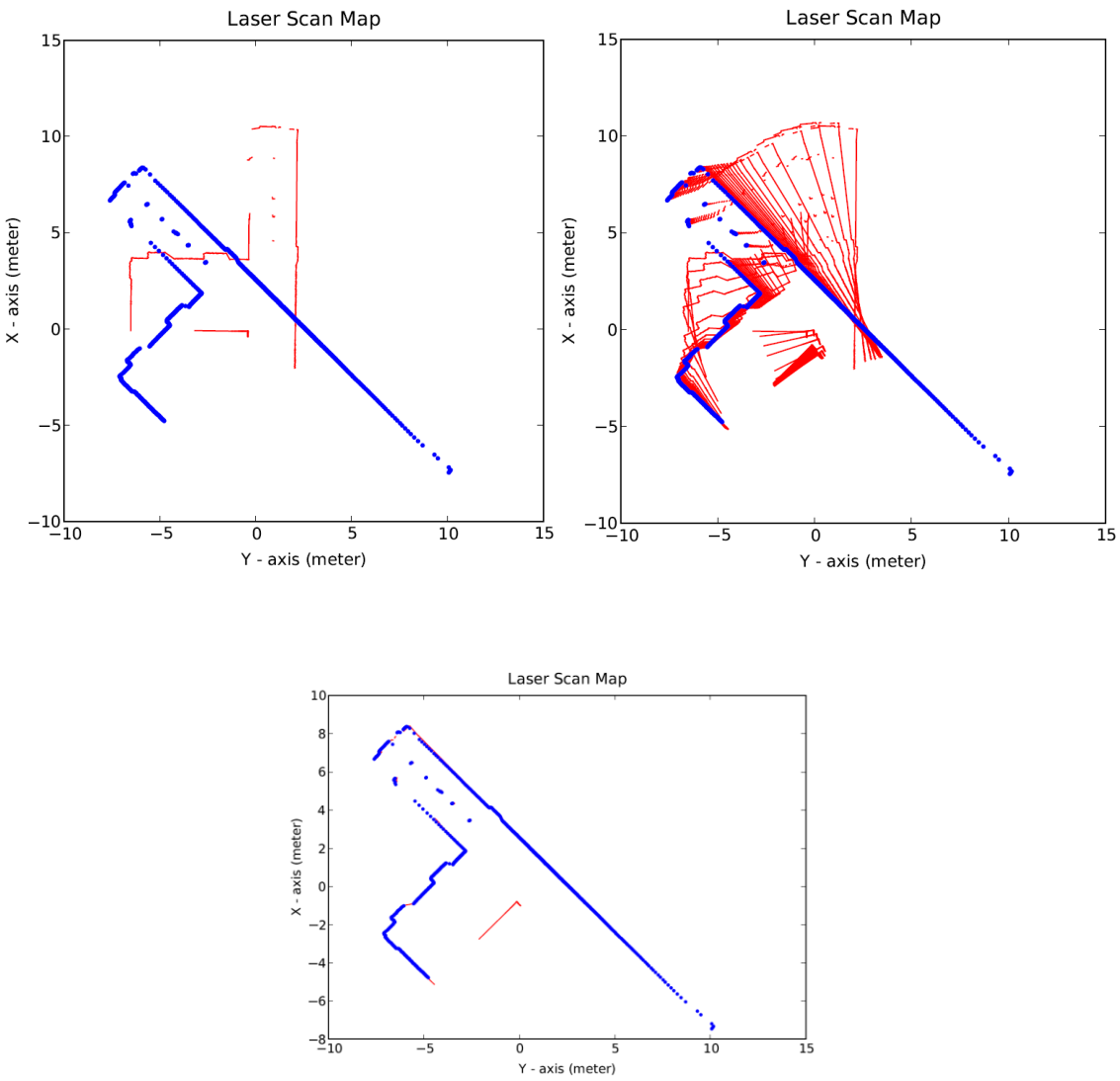


Figure 38. An example of the ICP scan-matching process that matches a line-based map with a point-based map. The map in blue represents the point-based map consisting of 1047 points. The map in red represents the line-based map consisting of 316 line segments. The upper-left figure shows the initial poses of the two maps. The upper-right figure shows all of the iterations for the line-based map to converge toward the point-based map. The lower figure shows the end of the last iteration of the convergence. The original scans were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

Table 2. A comparison in time consumption of the process while matching two different types of maps with a point-based map without applying down-sampling to the maps for the correspondence search of each ICP iteration. The raw measurements used for both setups are the same as the example shown in Figure 38. The program was implemented by using the Python script language and executed on a laptop with an Intel Core i7 2.20GHz processor. The code in the program is never optimized, and the value produced is only intended for giving a rough concept.

Scan-matching map type	# of Iterations	Time (Seconds)
Matching two point-based maps	19	10.7217815053
Matching a line-based map with a point-based map	23	7.81693578522
Threshold (Meters^2): Improvement of the latest iteration in reducing the mean squared distance between every pair of corresponding points <= 0.000025		

2. Line Feature-Based Transformation Correction

As discussed in Section E of this chapter, the issue associated with the point-based map-building and localization process is that the estimation error is cumulative. For the line-based process, the issue still exists since the same ICP algorithm is used. It is noted that adjusting certain threshold values for the process to take more ICP iterations can lead to a better scan-matching accuracy. In this way, the overall error accumulated during the process can be reduced to a certain degree. However, more iterations also cause greater time consumption but only provide limited improvement. Some other means should still be considered.

A mechanism that utilizes the line features (with the attributes such as the bearing θ and the range r) is proposed in this research to correct the transformation between two line-based maps. This is one of the reasons that the line-based representation for maps is adopted in this research.

Assuming two line segments A and B correspond to each other (as shown in Figure 39), to align the line extending segment A with the line extending segment B, there are two steps to be followed.

- Rotate segment A about the origin by an angle of $(\theta_2 - \theta_1)$.
- Translate segment A by $(x_0,\ y_0)$, where

$$x_0 = (r_2 - r_1)\cos\theta_2\ , \quad y_0 = (r_2 - r_1)\sin\theta_2. \tag{3.16}$$

The unit quaternion to perform the required rotation is given by

$$q_{(\theta_2-\theta_1)} = \left[\cos\left(\frac{(\theta_2-\theta_1)}{2}\right), 0, 0, \sin\left(\frac{(\theta_2-\theta_1)}{2}\right)\right]. \tag{3.17}$$

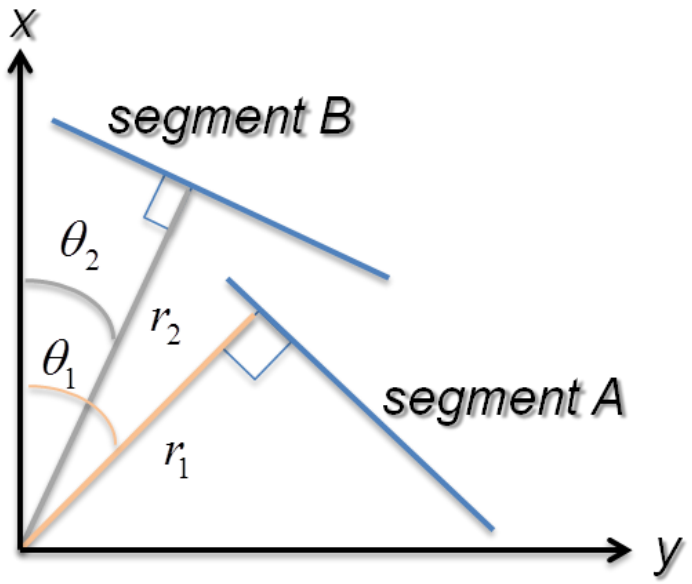


Figure 39. Two corresponding line segments in a x-y coordinate system.

To correct the transformation between two line-based scans, the line feature-based correction mechanism can be implemented by aligning the pair of corresponding line segments determined from the two separate scans. For a better alignment, the same process needs to be completed for two or more times using different corresponding pairs that are not in parallel with one another. This method assumes that the corresponding pair between two scans can be found. However, determining the correspondence correctly is difficult, and adopting an incorrect one will only worsen the original alignment error between the two scans. This is the reason that the correction mechanism will be built as an extension of the ICP algorithm. While the two scans are nearly matched after the completion of ICP algorithm, the corresponding pairs can be determined by justifying the differences in the line features of the two scans, such as the lengths, the ranges, the bearings, and the mid-point positions. The estimation error for the transformation between the two scans can thus be reduced.

The determination of corresponding pairs between two line-based scans (for example, scans X and Y) in the line feature-based transformation correction mechanism is conducted by completing the following steps.

- Extract the 10 longest line segments from scan X.
- For each of the 10 longest line segments in scan X, extract the line segment from scan Y that has the smallest length difference as its corresponding line segment.
- Verify each of the 10 corresponding pairs by examining whether the following conditions are met.
    - Both of the corresponding line segments are longer than 2 meters.
    - The difference between the lengths of the corresponding line segments is smaller than 1 meter.
    - The difference between the ranges (i.e., r_1 and r_2 as shown in Figure 39) of the corresponding line segments is smaller than 1 meter.
    - The difference between the bearings (i.e., θ_1 and θ_2 as shown in Figure 39) of the corresponding line segments is smaller than 0.4 radians.
    - The distance between the mid-points of the corresponding line segments is shorter than 0.5 meters.

Note that it is possible that no corresponding pair is found. The number of line segments to be extracted and the threshold values can be adjusted to meet the needs of matching difference scans. The values adopted in this dissertation are solely intended for examining the feasibility of the transformation correction mechanism. They may not necessarily be optimal.

An example of the result of combining the ICP algorithm with the line feature-based transformation correction mechanism is shown in Figure 40. The upper figure and lower figure show the results before and after using the correction mechanism, respectively. One pair of the corresponding line segments is determined. The small alignment error between the red scan and the blue scan in the upper figure can be corrected by applying the line feature-based transformation correction mechanism. Note that before the line feature-based transformation correction mechanism is implemented, the point-based map needs to be converted into a line-based map. The lower figure that

shows the convergence of the line-based map and the point-based map instead of two line-based maps is only intended for an easier comparison with the upper figure.

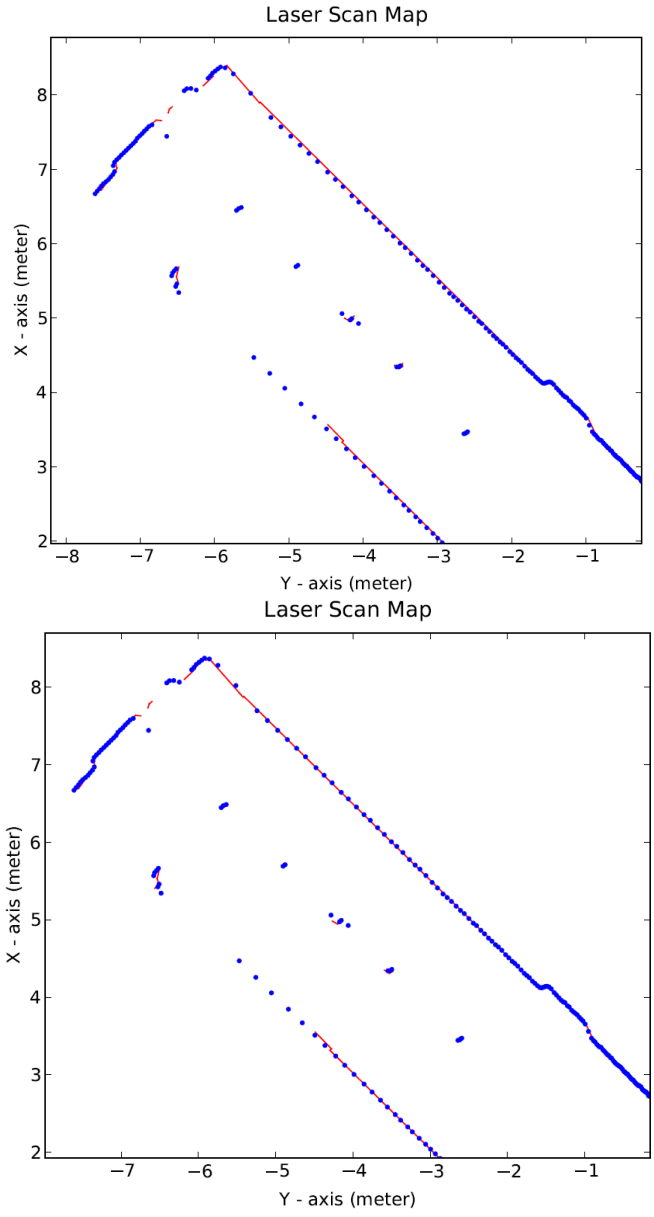


Figure 40. An example of the result of combining the ICP algorithm with the line feature-based transformation correction mechanism. The upper figure and lower figure show the results before and after using the correction mechanism, respectively. The small alignment error between the red scan and the blue scan in the upper figure can be corrected by applying the line feature-based transformation correction mechanism. The original scans were taken in Spanagel Hall (5th floor corridor near the east-side elevators) of the Naval Postgraduate School.

3. Real-time Map-building and Localization using the Combination of the ICP Algorithm and the Line Feature-based Transformation Correction Mechanism

The incremental process for line-based map-building and localization in this research is to build and maintain a line-based map while simultaneously localizing the ranging measurement system. This process begins with an initial point-based scan that is then converted into a line-based one and treated as the original old scan. (Note that a known map built beforehand can also be adopted as the original old scan, as long as it can be converted into a line-based map.) For every subsequent new point-based scan acquired, the process cycle completes the following steps.

- Match the old line-based scan with the newly acquired point-based scan using the ICP algorithm.
- Convert the newly acquired point-based scan into a line-based scan.
- Correct the transformation of the transformed old line-based scan to better match with the new one using the line feature-based transformation correction mechanism if the corresponding line segments among the two scans can be determined.
- Merge the two matched line-based scans into a single line-based scan using the same concept as adopted in Section D of this chapter for merging two point-based scans. The resulting line-based scan is regarded as the newer old scan for the next process cycle.

The estimation for the transformation (i.e., the orientation and position) of the ranging measurement system at the end of each process cycle can be implemented using the same formula as discussed in Section E (Equation 3.10) of this chapter.

An example of the post-processed line-based map-building and localization is shown in Figure 41. It is intended for investigating the performance in terms of estimation accuracy in comparison with the point-based process as shown in Figure 33. The process adopts the same ICP threshold value and is fed with the same raw sensor data. As mentioned earlier, it is difficult to tell the estimation error from the final map. Instead, it can be reflected by the estimated trajectory of the ranging measurement system. Judging by the distances between the estimated initial and final positions for the point-based and line-based processes, the line-based process proposed in this section performs relatively better because the accumulated estimation error can be reduced.

Since the line-based map-building and localization process is relatively more efficient than the point-based process, there is some leeway in trading efficiency for higher accuracy. By adopting a lower threshold value for the ICP algorithm, the process can reach higher accuracy and still be suitable for the real-time implementation. An example of the process applied with such adjustment and fed with the same raw sensor data used in the previous example is shown in Figure 42. It is noted that the estimation accuracy is enhanced since the distance between the estimated initial and final positions is reduced.

An example of the real-time line-based map-building and localization process is shown in Figure 43. To examine the estimation accuracy of the process, an infrared precision position tracker (PPT) system[2] [61] is used to concurrently estimate the position of the ranging measurement system. The PPT measurement was reported to have an overall root-mean-square (RMS) error in centimeters (6.68 cm, to be exact) and thus is regarded as the ground truth. A comparison of the trajectories of ranging measurement system obtained by using both the real-time, line-based map-building and localization process and the PPT system is shown in Figure 44. The comparison indicates that the proposed process offers high accuracy, since the two trajectories closely coincide with each other. (In this example, the RMS error of the trajectory determined by the real-time, line-based map-building and localization process relative to the one determined by the PPT system is approximately 0.07 meters.) Note that the points in time at which the measurements are taken for both approaches are not necessarily the same. This is the reason that the points from the two trajectories do not sit exactly on top of each other.

[2] The test facility used is an indoor basketball court, which is also the huge immersive virtual environment (HIVE) equipped with an infrared precision position tracker (PPT) system at Miami University in Oxford, Ohio.

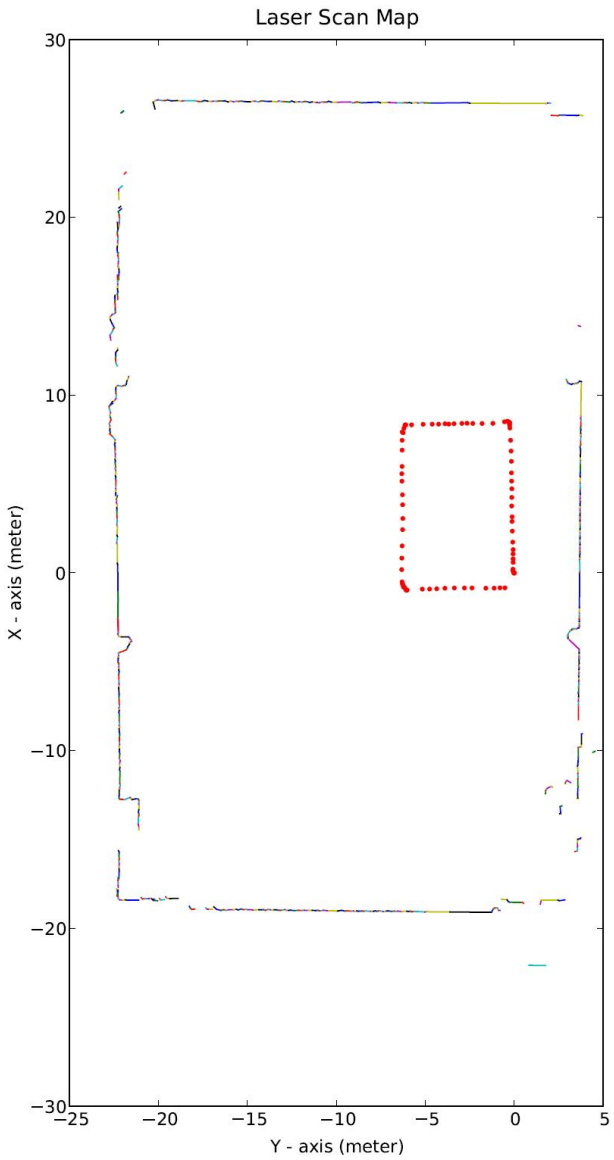


Figure 41. An example of the line-based map-building and localization process using the combination of the ICP algorithm and the line feature-based transformation correction mechanism. To show the performance in terms of estimation accuracy in comparison with the point-based process as shown in Figure 33, a post process with the same threshold value for the ICP algorithm is implemented and fed with the same raw sensor data. The line set in different colors represents the resulting map of the environment, while the point set in red shows the trajectory of the ranging measurement system. The map consisting of 648 line segments was built by matching and merging 81 individual scans. Although the actual initial and final positions were both (0, 0), the estimation result shows that the initial position (0, 0) and the final position (-0.841606400882, - 0.528697346803) do not match due to the error accumulated during each of the scan-matching processes. The original test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

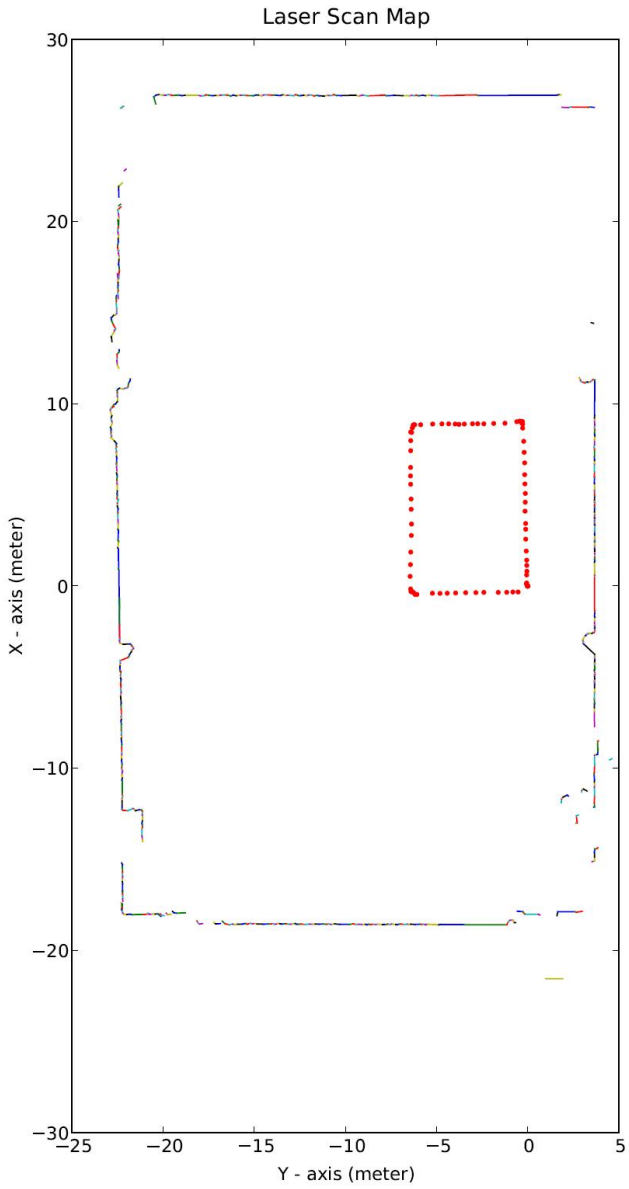


Figure 42. An example of the line-based map-building and localization process using the combination of the ICP algorithm and the line feature-based transformation correction mechanism. The same raw sensor data used in the example shown in Figure 41 are adopted. A lower threshold value of the improvement in reducing the mean squared distance between every pair of corresponding points for the ICP algorithm is adopted for a higher performance in terms of estimation accuracy. The line set in different colors represents the resulting map of the environment, while the point set in red shows the trajectory of the ranging measurement system. The map consisting of 648 line segments was built by matching and merging 81 individual scans. The actual initial and final positions are both (0, 0). The estimates for initial and final positions are (0, 0) and (-0.327111495504, -0.533314961374), respectively. The original test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

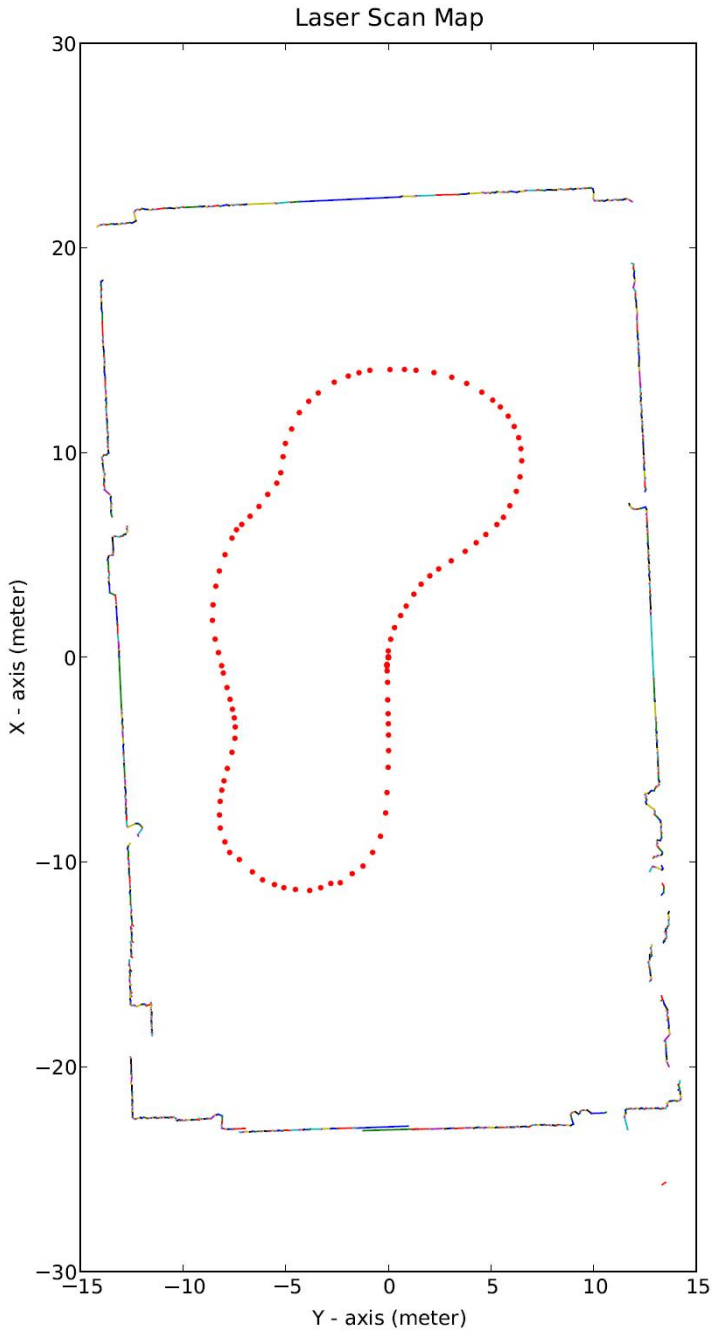


Figure 43. An example of the real-time line-based map-building and localization process using the combination of the ICP algorithm and the line feature-based transformation correction mechanism. The line set in different colors represents the resulting map of the environment, while the point set in red shows the trajectory of the ranging measurement system. The map consisting of 846 line segments was built by matching and merging 106 individual scans. The actual initial and final positions are both (0, 0). The estimates for initial and final positions are (0, 0) and (-0.348965542645, -0.071385533758), respectively. The test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

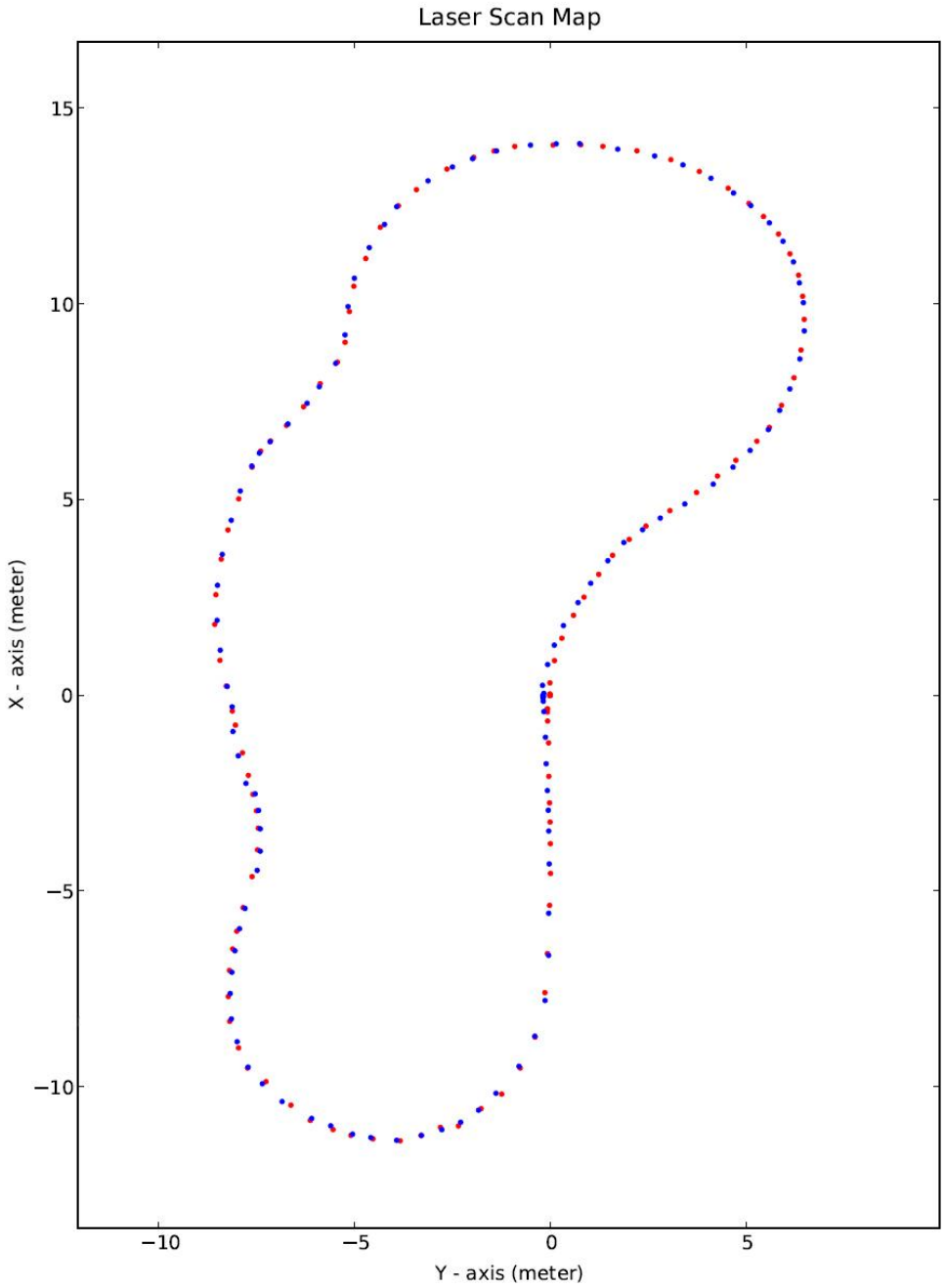


Figure 44. A comparison of the trajectories of ranging measurement system. The trajectory in red is estimated by implementing the real-time line-based map-building and localization process, while the trajectory in blue is estimated by using the infrared precision position tracker (PPT) of the huge immersive virtual environment (HIVE) [61]. The HIVE is located in an indoor basketball court of Miami University in Oxford, Ohio. The RMS error of the trajectory determined by the real-time, line-based map-building and localization process relative to the one determined by the PPT system is approximately 0.07 meters.

H. LOCAL MINIMUM ESCAPE MECHANISM FOR THE ICP ALGORITHM

As discussed in Section C of this chapter, the ICP algorithm almost always guarantees bringing two scans closer to a local minimum. This local minimum may ideally be the global minimum most of the time. It is noted that rejecting a portion of outliers can effectively prevent the process from being trapped in one of the many local minima. However, even when the outlier rejection mechanism is applied, the local minimum issue can still occur occasionally. As shown in Figure 45, when the process is trapped in a local minimum, the two scans will never coincide with each other no matter how many ICP iterations the process takes.

There are various techniques that approach the optimization (maximization or minimization) problem in the literature. Most of them need to face the similar local optimum issue.

For example, a greedy-based algorithm, such as hill-climbing [62], is a mathematical optimization technique that begins with an (arbitrary) solution to a problem and iteratively finds a better solution by incrementally making a small change to the solution. The ICP algorithm used in our research is similar to the greedy-based algorithm, which is usually efficient and guarantees finding a local optimum. However, this also raises an issue because the local optimum is not necessarily the global optimum.

Simulated annealing (SA) [63], with its idea inspired by the annealing process in metallurgy, is used to determine an acceptable approximation of the global optimum within a given fixed amount of time. With this technique, the process is able to escape the local optimum by allowing some solutions that are equal to or worse than the current solution but gradually decreasing the probability of allowing them as the process goes toward the finish time.

It is also possible for the greedy-based algorithm to escape the local optimum. There are generally two methods:

- Specify a random initial condition for a process restart.
- Apply a random amount of sideway move.

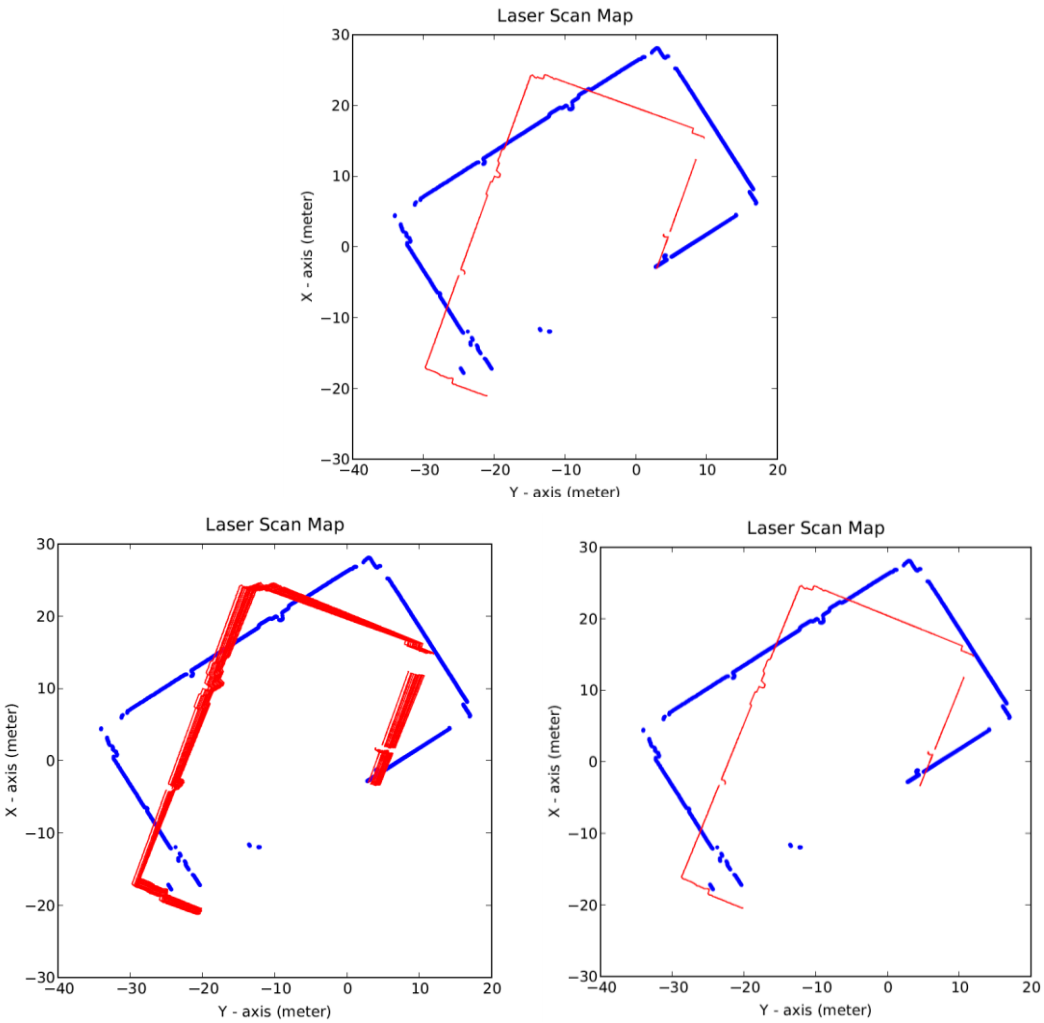


Figure 45. An example of the ICP scan-matching process that is trapped in a local minimum. The process matches a line-based scan with a point-based scan. The scan in blue represents the point-based scan consisting of 1014 points. The scan in red represents the line-based scan consisting of 488 line segments. The upper figure shows two individual scans. The lower left figure shows all of the iterations for the line-based scan to converge toward the point-based scan. The lower right figure shows the end of the last iteration of the convergence. The two scans were taken in an indoor basketball court of Miami University in Oxford, Ohio.

Both of the methods are suitable for use in the ICP process. A random (or user-specified) amount of rotation serving as the combination of the two methods can be applied for the ICP process to escape the current local minimum. To incorporate this mechanism in the ICP algorithm, the process cycles through the following steps:

- Implement the ICP iterations until the threshold is reached.
- Determine whether the process is trapped in a local minimum by examining the mean distance between every pair of corresponding points.
- Apply a random (or user-specified) rotation to the scan being matched with the other scan if the process is trapped in a local minimum. The direction of such rotation will be the same as the rotation during the previous ICP iteration.
- Restart the process and implement the ICP iterations until the threshold is reached.

Figure 46 shows the result of the ICP scan-matching process combined with the local minimum escape mechanism for matching the same scans shown in Figure 45.

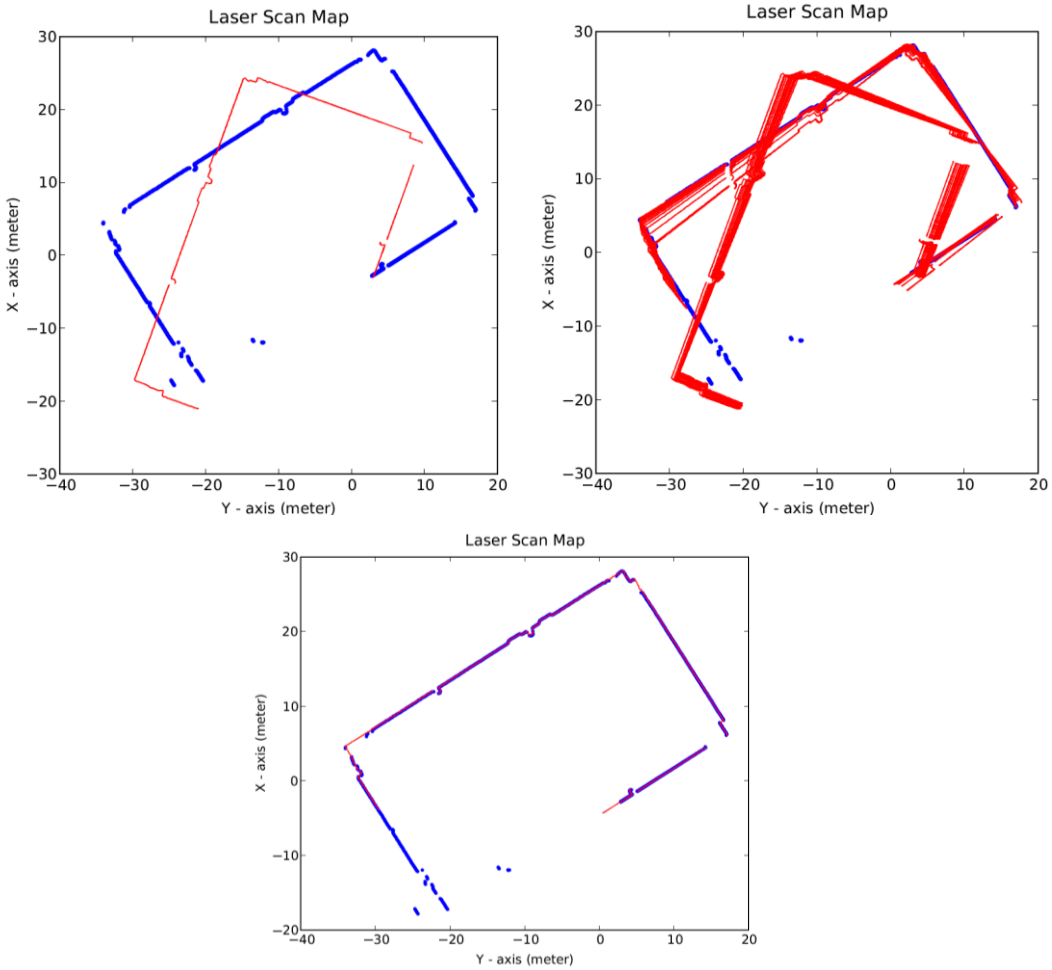


Figure 46. An example of the ICP scan-matching process combined with the local minimum escape mechanism. The process matches a line-based scan with a point-based scan. The scan in blue represents the point-based scan consisting of 1014 points. The scan in red represents the line-based scan consisting of 488 line segments. The upper left figure shows two individual scans. The upper right figure shows all of the iterations for the line-based scan to converge toward the point-based scan. The lower figure shows the end of the last iteration of the convergence. The two scans were taken in an indoor basketball court of Miami University in Oxford, Ohio.

I. LIMITATIONS

The ranging measurement system-based map-building and localization process proposed in this chapter can be effectively implemented under the following assumptions and limitations:

- Between every two consecutive scans, a sufficient portion must overlap to guarantee successful scan matching. This offers the scan-matching process

enough correspondence information between the two scans being matched. In other words, the transformation between every two consecutive scans is assumed to be small. For instance, it is recommended that the magnitude of rotation be less than 45 degrees, and that the norm of translation be less than 10 meters.

- Partly following the previous assumption, the velocity of the ranging measurement system during the implementation of the process needs to be as low as humans' normal walking speed to keep the transformation between two consecutive scans small. In addition, it takes some duration of time for the ranging measurement system to measure the ranges from its environment in a counter-clockwise manner. The scan will be skewed if the ranging measurement system is moving during scanning. This is another reason that the velocity needs to be kept low to reduce the skewing effect.
- The dimensions of the environment need to be in accordance with the effective range of the ranging measurement system. For a large open space with boundaries beyond the effective range of the system, there will be little or no correspondence between scans for matching, and thus it is not suitable for the use of our process. In addition, a space that is narrow and long, such as a corridor, is not suitable either. The only features that the process is able to determine from the scans are two parallel walls. Both ends of the corridor are invisible due to the insufficient effective range of the system and the large incident angles of the laser or ultrasonic waves. Under this circumstance, it is difficult for the process to estimate the relative translation between scans. An example of the scan-matching process for such a corridor environment is shown in Figure 47. The actual environment for this particular example is shown in Figure 48.

There are certain conditions under which the process does not operate well because the first assumption is not met, even when the transformation between two consecutive scans is small. For example, when the ranging measurement system moves from one space into another through a small passage (e.g., a doorway), the scans taken before and after passing through the doorway may contain little correspondence and thus are not able to be properly matched. A similar situation may occur when moving around a corner, because there may be too much new information being gathered around the corner relative to the old information in the newly acquired scan.

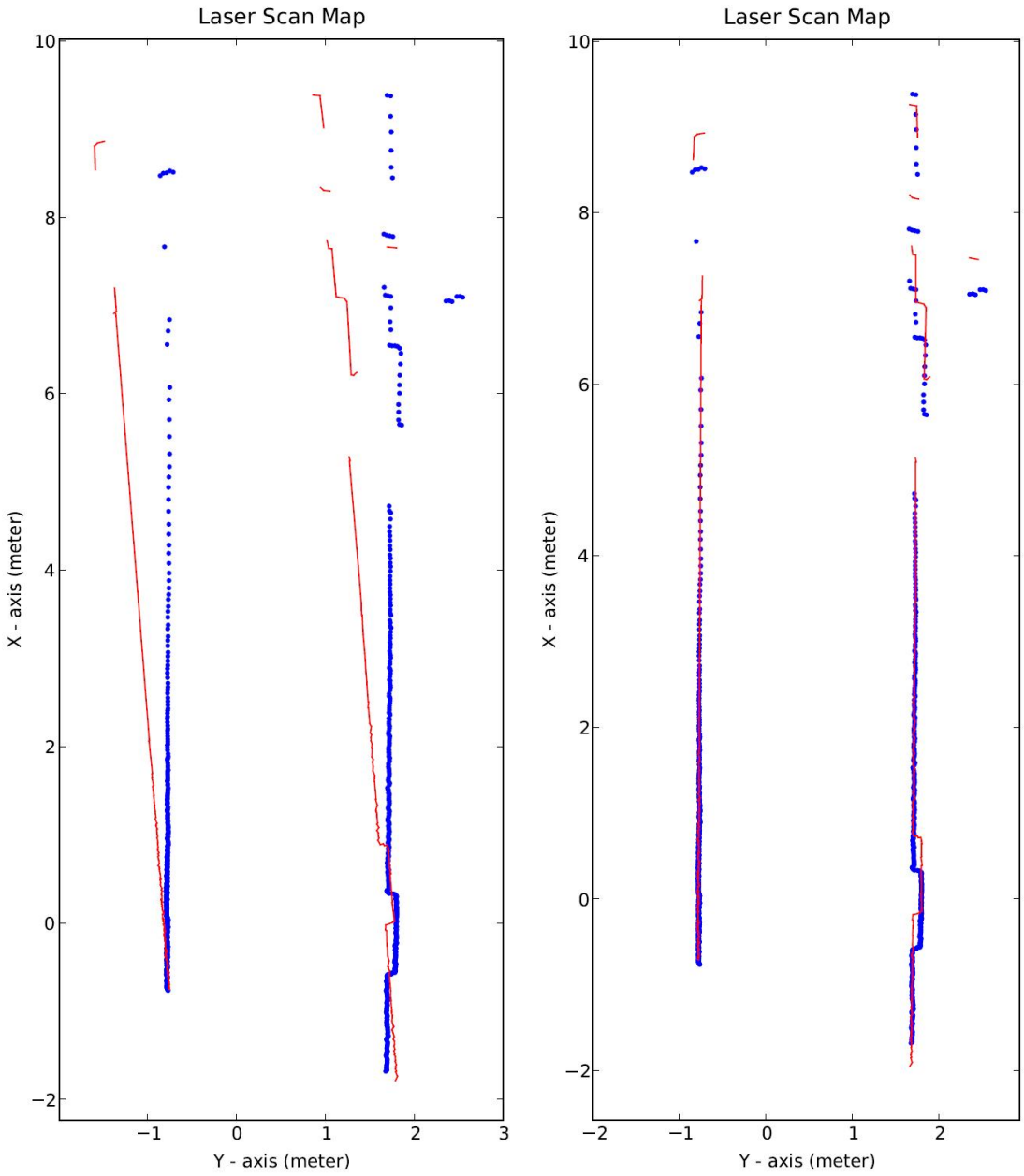


Figure 47. An example of the scan-matching process for a corridor environment. The process matches a line-based scan with a point-based scan. The scan in blue represents the point-based scan consisting of 1035 points. The scan in red represents the line-based scan consisting of 252 line segments. The left figure shows two individual scans before being matched, while the right figure shows the end of the last iteration of the convergence. It is noted that the scan-matching result has large error along the direction parallel to the walls. The scans were taken in Spanagel Hall (5th floor corridor) of the Naval Postgraduate School.

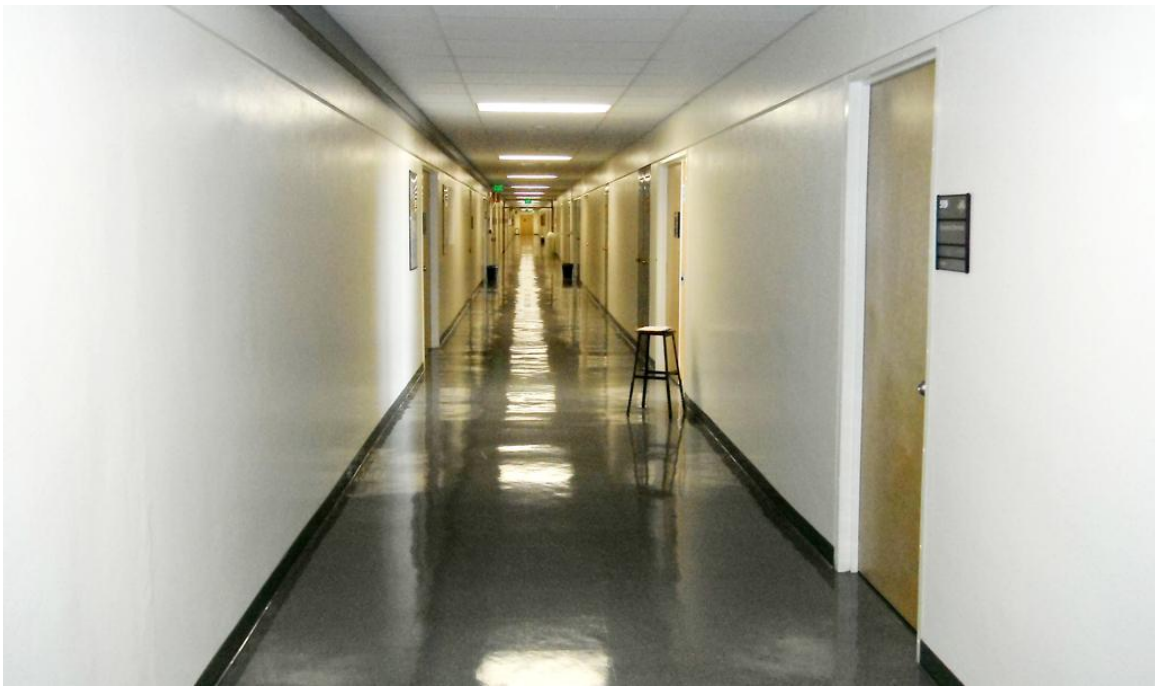


Figure 48. The actual environment for the example as shown in Figure 47. The picture was taken in Spanagel Hall (5th floor corridor) of the Naval Postgraduate School.

J. SUMMARY

This chapter presented a ranging measurement system-based map-building and localization algorithm. It started with the investigation of the point-based map-building and localization process using the well-known ICP algorithm. The ICP algorithm always guarantees bringing two scans being matched closer toward a local minimum. However, the local minimum is not necessarily the global minimum being sought. To deal with the local minimum issue often caused by the outliers, a mechanism that rejects a portion of the outliers was incorporated during each of the ICP iterations. For the real-time implementation, a tree-search technique and a down-sampling mechanism were adopted for executing the correspondence search to reduce the time consumption of the process.

Although the point-based map-building and localization process is able to produce acceptable results, there are still efficiency and accuracy issues that need to be addressed. The line-based representation for the map being built was adopted to reduce the time consumption of the process. Furthermore, a line feature-based transformation correction mechanism was constructed to correct the alignment error between two matched scans.

The resulting line-based map-building and localization process performs relatively better than the original point-based process.

Finally, the approaches to escaping the local minima were discussed. Although the local minimum issue can be resolved to a certain degree by rejecting some of the outliers, the scan-matching process can still be trapped in one of the local minima occasionally. Therefore, a local minimum escape mechanism was incorporated in the process to further resolve this issue.

The main contributions of this chapter are listed as follows:

- An investigation was conducted to present the performance difference between the point-based and line-based map-building and localization process using the ICP algorithm.
- An outlier rejection mechanism was incorporated in the ICP algorithm to improve the scan-matching result so that the local minimum issue caused by the outliers can be resolved to a certain degree.
- A mechanism that utilizes the Hough transform was constructed to approximate the original point-based map by a set of line segments.
- A line feature-based transformation correction mechanism was constructed to correct the alignment error between two matched scans.
- A local minimum escape mechanism was incorporated in the process to further resolve the local minimum issue.
- A real-time, line-based map-building and localization process was constructed by combining the ICP algorithm with the line feature-based transformation correction mechanism and the local minimum escape mechanism.

IV. IMPROVED MAP-BUILDING AND LOCALIZATION ALGORITHMS FOR HIVE ENVIRONMENT

A. SEPARATION OF MAP-BUILDING AND LOCALIZATION PROCESSES

1. Motivation

In the previous chapter, a real-time, line-based map-building and localization algorithm was constructed by combining the ICP algorithm with the line feature-based transformation correction mechanism and the local minimum escape mechanism. This algorithm is suitable for the use of simultaneous localization and map-building (SLAM). However, there are some particular cases, such as the huge immersive virtual environment (HIVE), in which the environment is merely a rectangular-shaped space with its longest side shorter than the maximum effective range of the ranging measurement system. Under this circumstance, the process for building the map is no longer required all the time as long as the map being built is roughly complete. In other words, map-building and localization do not necessarily need to be implemented simultaneously. The original process can be divided into two separated ones. The map-building process is implemented beforehand to build a map of the environment, while the localization process that determines the whereabouts of the ranging measurement system within this map becomes the main focus after the map-building process is completed.

Furthermore, the estimation error for the original process using the presented algorithm may still be accumulated over time despite the fact that it has been reduced, as discussed in the previous chapter. This issue is expected to be resolved by using the separated processes since the estimation error will not be further accumulated after the map-building process is completed. The estimation error for the subsequent localization process can thus be bound within a certain range, because the process becomes restrictedly the scan-matching process between every newly acquired scan and the fixed map, and the error of every scan-matching process is independent and is not cumulative. The value of this bound error will correlate closely with the overall error accumulated during the map-building process. The localization process is expected to offer high accuracy when a well-built map is used.

2. Environment Requirements

To apply the concept of the two separated processes, it is necessary to ensure that the map can be built accurately, so that the subsequent localization process can be implemented properly. This is accomplished by following the two requirements when adopting the operating environment.

- The environment needs to meet the third assumption discussed in Section I of Chapter III for the original ranging measurement system-based map-building and localization process to properly operate.
- The environment needs to be a convex polygonal shape. This is required for the ranging measurement system to detect most of the boundaries directly without obstruction at any position within the environment. A simpler environment such as a general rectangular-shaped indoor space is recommended for obtaining an accurate result.

B. MAP-BUILDING PROCESS

The separated map-building process is intended to build the map of the operating environment for later use in localization. It only needs to be completed once for the same environment. The algorithm used for this process is the same as the original line-based map-building and localization process presented in the previous chapter with some modifications made. Three major modifications are summarized as follows.

- An offline process approach is adopted instead of a real-time approach. This is intended to speed up the scanning rate, since the process does not spend time completing scan matching and merging during the period in which the scans are taken. Given a higher scanning rate, the transformation between every two consecutive scans is thus smaller, and a better scan-matching result may be produced. This concept is similar to the first assumption discussed in Section I of Chapter III. However, the scanning rate is adjusted to be as low as 2 Hz in this chapter to reduce the number of scans that need to be processed. The first stage of this map-building process is to take scans that cover most boundaries of the environment. At this stage, it is necessary to move the ranging measurement system around to ensure complete scanning coverage but also to keep the trip short to avoid unnecessarily accumulating the estimation error. A small circle around the environment is suggested. The second stage is to process the scans and to build the map off line by using the same algorithm as that used in the original line-based map-building and localization process.
- The merging mechanism between two matched scans used in this separated process keeps more information than the one used in the original

algorithm. The original merging mechanism discards a larger portion of map information in exchange for a higher efficiency. However, building a more detailed map is the main focus of this separated process, and the efficiency for merging the scans is no longer an issue since this process is implemented off line. An additional correspondence search between the old and new scans is conducted to determine whether the information in the old scan needs to be discarded. Only the line segments of the old map with their correspondence found within a given range are regarded as the duplicate information and are discarded.

- The merging mechanism is not implemented every process cycle. Following the previous modification, the amount of information in the map may become massive as the number of scans being merged grows. This may affect the efficiency of the later uses of the localization process. To reduce the amount of information but still keep necessary details in the final map, only a given number of scans are merged.

An example of the separated map-building process is shown in Figure 49. The line set in different colors represents the final map of the environment, while the point set in red shows the trajectory of the ranging measurement system as the scans were taken. The trajectory for this process is arbitrary as long as the scans contain most boundaries of the environment. In this particular example, the ranging measurement system was moved around roughly in a small circle to ensure that sufficient information of this environment was gathered. Figure 50 shows the magnified plot of the trajectory of the ranging measurement system.

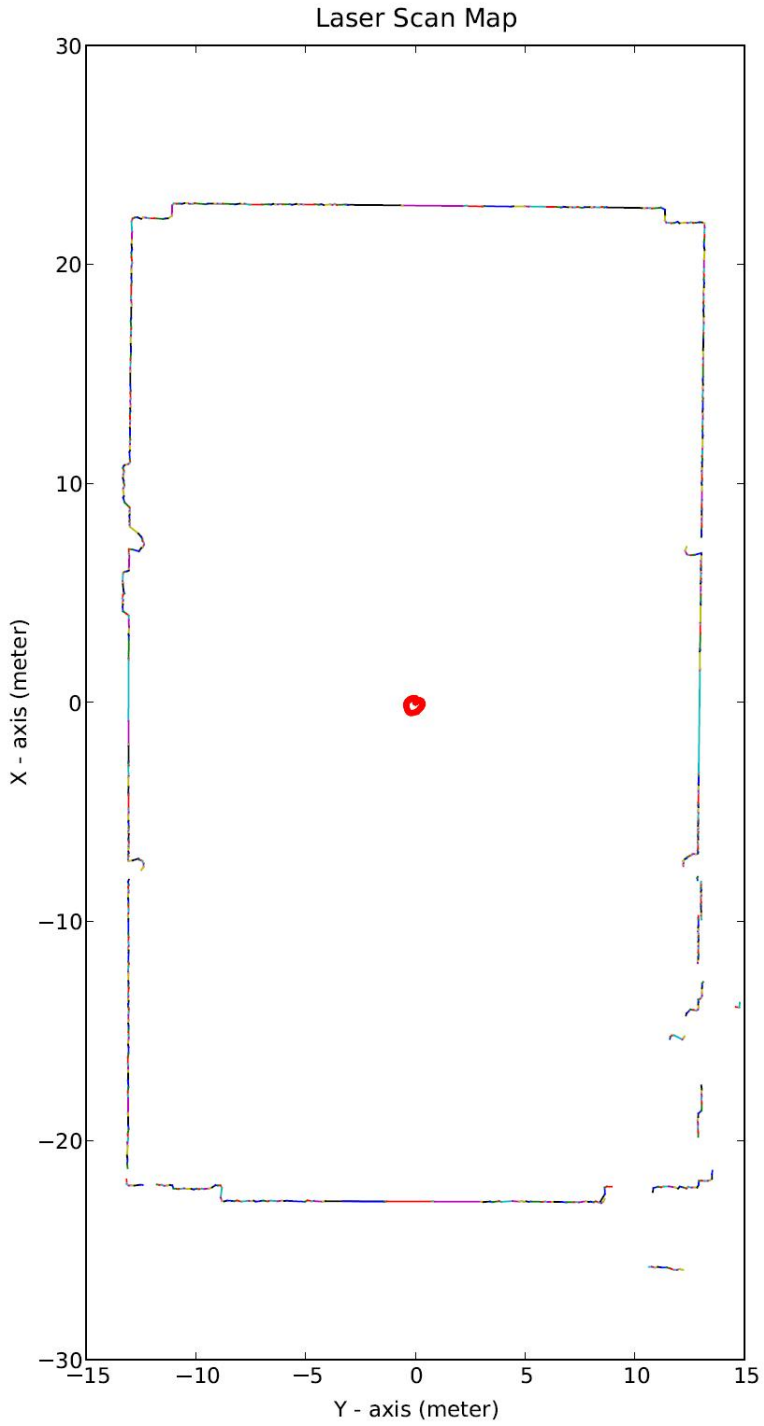


Figure 49. An example of the separated map-building process. The line set in different colors represents the resulting map of the environment, while the point set in red shows the trajectory of the ranging measurement system. The map consisting of 713 line segments was built by matching 119 individual scans but merging only 10 of them. The 10 merged scans were chosen evenly throughout all of the scans. The test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

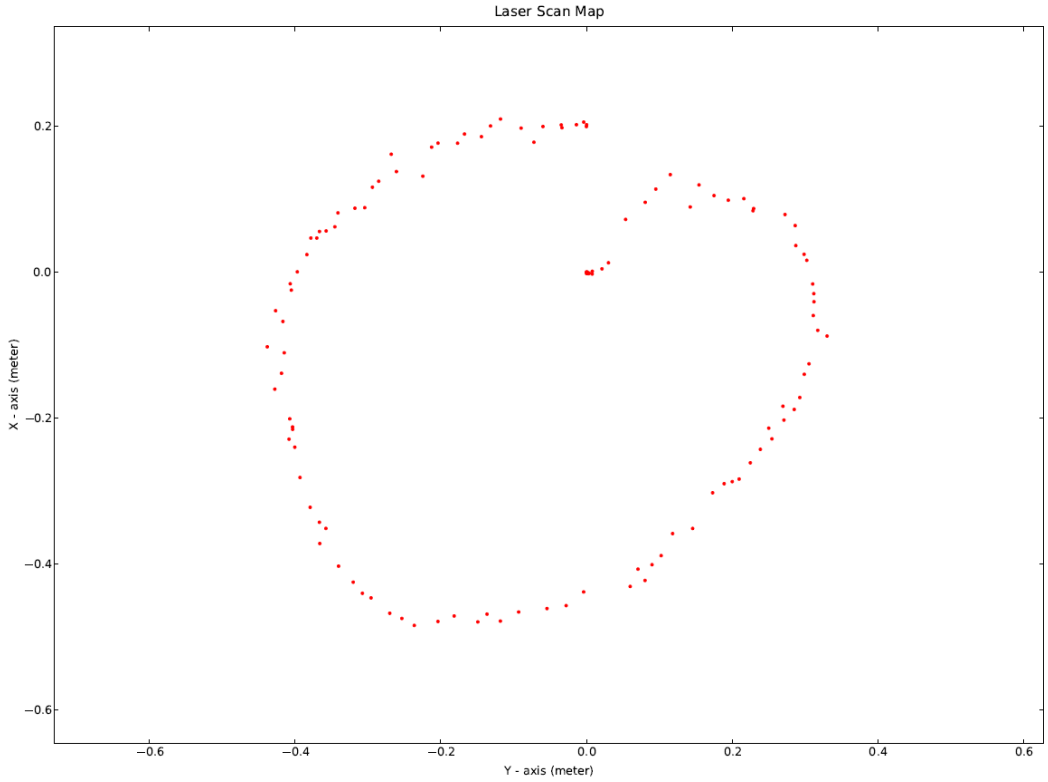


Figure 50. Trajectory of the ranging measurement system when building the map as shown in Figure 49. The start and end positions are (0.0, 0.0) and (0.199130768983, -0.000033682402), respectively.

As discussed in the previous chapter (Section I), a space that is narrow and long, such as a corridor, is not suitable for the original map-building and localization process. It may not be suitable for the separated map-building process either. However, it is worth indicating the improvement obtained by using the separated map-building process in comparison with the original map-building and localization process in coping with such an environment. The scan-matching error can be reduced to a certain degree. The results for the same range data processed by the two processes are shown in Figure 51. The scans were taken from one end of a corridor to the other. The map built by using the separated map-building process better represents the shape of the actual corridor environment, while the one obtained using the original map-building and localization process appears shorter and more crooked.

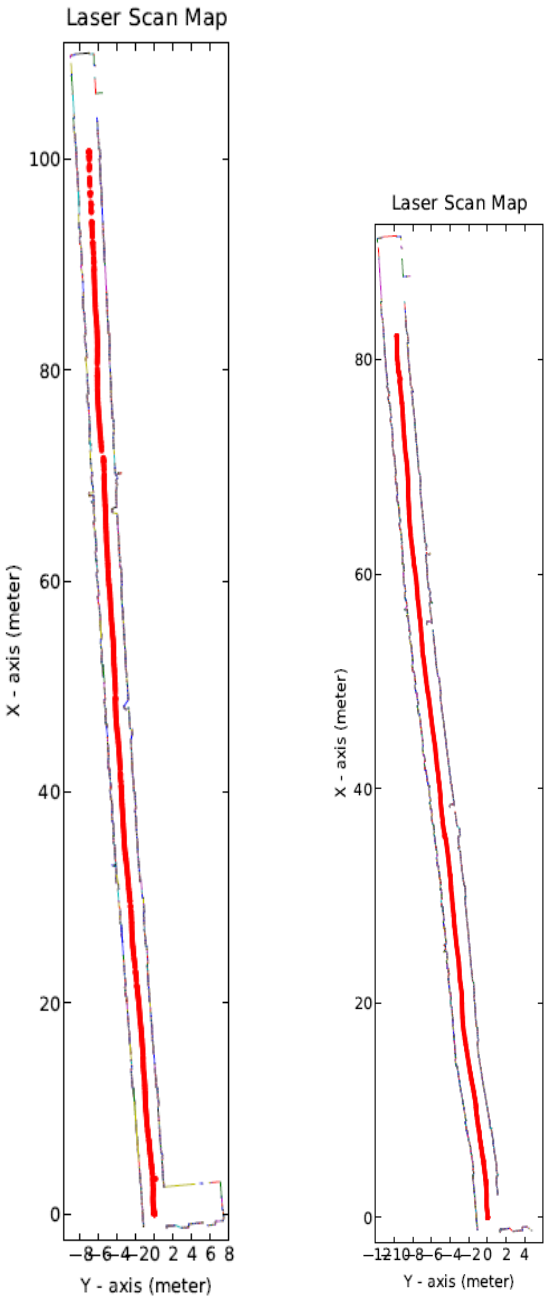


Figure 51. The comparison between the results of using the separated map-building process (left) and the original map-building and localization process (right) in a corridor environment. The final map on the left consisting of 5033 line segments was obtained by matching 862 scans but merging only 29 of them. The final map on the right consisting of 6295 line segments was obtained by matching and merging 862 scans. The point sets in red represent the estimated trajectories when collecting the range data. The map on the left better represents the shape of the actual environment, while the one on the right appears shorter and more crooked. The scans were taken in Spanagel Hall (5th floor corridor from one end to the other) of the Naval Postgraduate School.

C. LOCALIZATION PROCESS

The separated localization process is intended to determine the orientation and position of the ranging measurement system by matching the map with every newly acquired scan. The only difference of this separated process from the original real-time, line-based map-building and localization process is that the map is no longer being updated. In other words, the map is matched but not merged with every newly acquired scan. The map is the one built beforehand using the separated map-building process presented in the previous section.

Figure 52 shows an example of the separated real-time localization process using the map built in the example shown in Figure 49. The ranging measurement system was moved roughly in a rectangular loop (i.e., moved alternately between moving forward and turning right until back to the start position). For the example shown in Figure 52, a comparison of the trajectories of the ranging measurement system obtained by using both the separated real-time localization process and the precision position tracker (PPT) system is shown in Figure 53. It is noted that the trajectory obtained by the separated localization process closely matches the one obtained by the PPT system. (In this example, the RMS error of the trajectory determined by the separated real-time localization process relative to the one determined by the PPT system is approximately 0.02 meters.) The proposed process is able to offer high localization accuracy.

Another example of the separated real-time localization process is shown in Figure 54. The ranging measurement system was moved randomly from and back to the origin within the map.

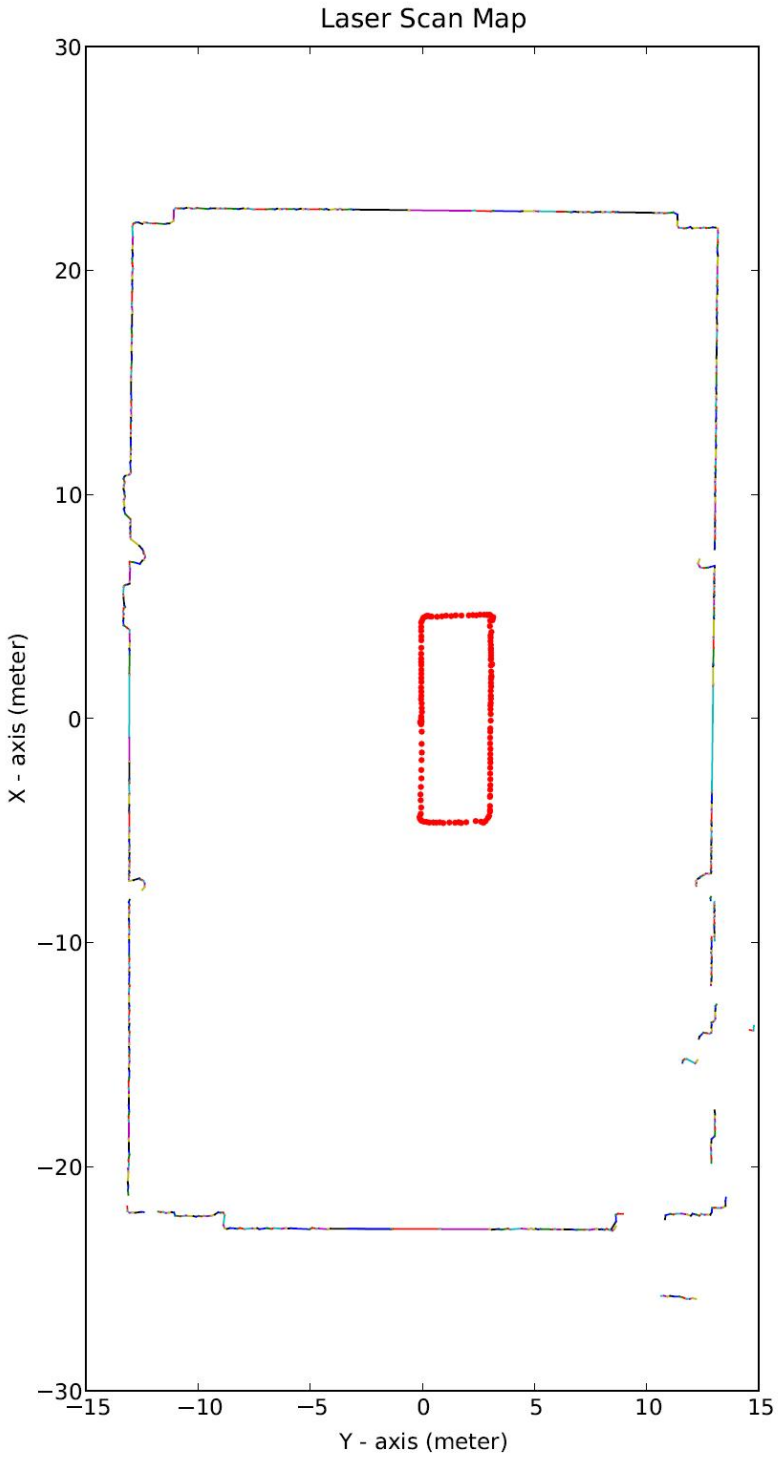


Figure 52. An example of the separated real-time localization process using the map built in the example as shown in Figure 49. The point set in red shows the trajectory of the ranging measurement system within the map. The map was matched against 138 new scans for estimating 138 data points within the trajectory. The test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

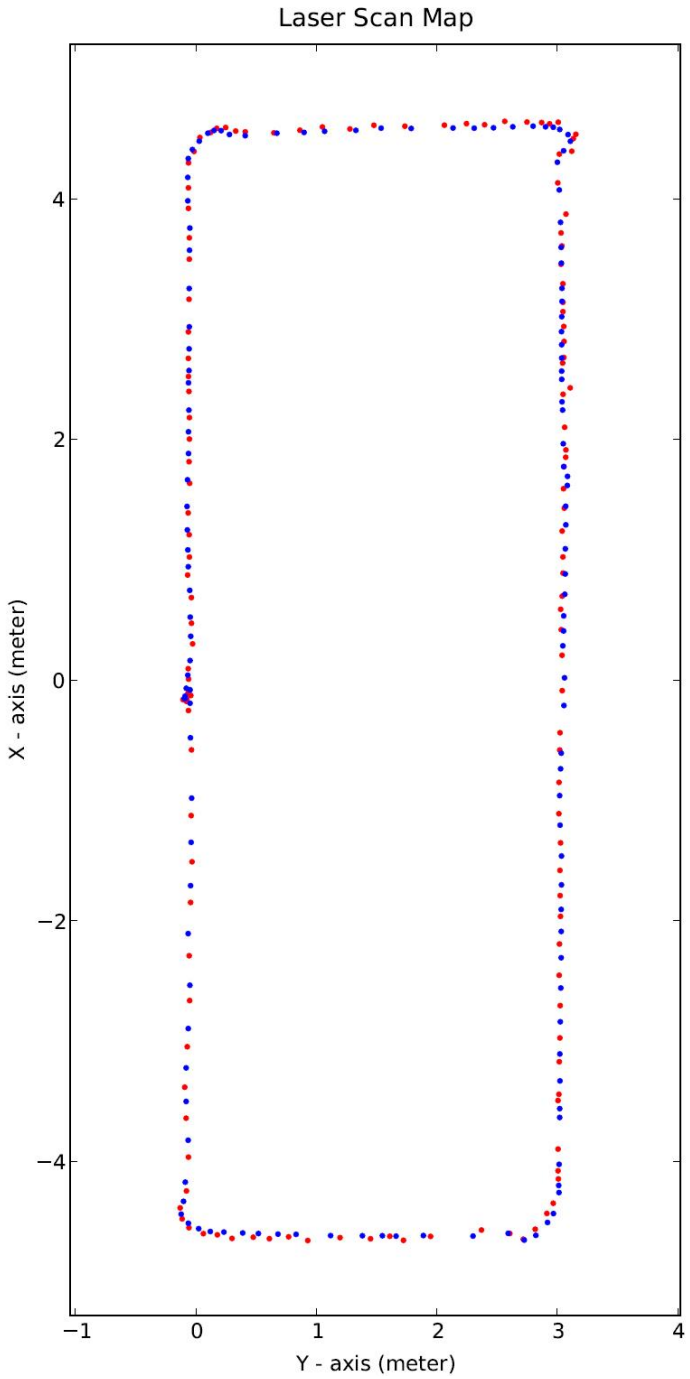


Figure 53. A comparison of the trajectories of ranging measurement system obtained by using both the real-time localization process and the PPT system. The trajectory in red is estimated by implementing the separated real-time localization process, while the trajectory in blue is estimated by using the PPT of the HIVE. The HIVE is located in an indoor basketball court of Miami University in Oxford, Ohio. The RMS error of the trajectory determined by the separated real-time localization process relative to the one determined by the PPT system is approximately 0.02 meters.

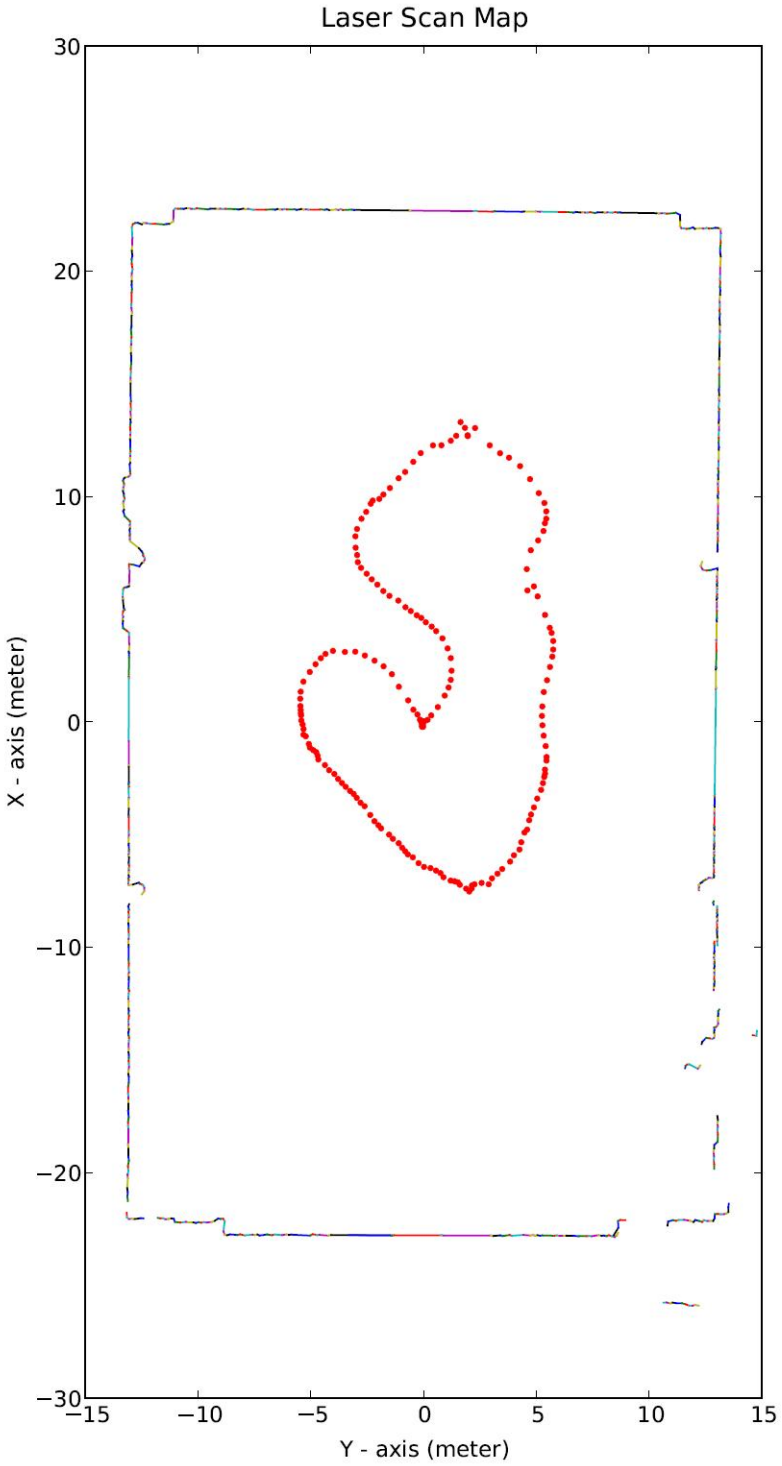


Figure 54. An example of the separated real-time localization process using the map built in the example shown in Figure 49. The point set in red shows the trajectory of the ranging measurement system within the map. The map was matched against 183 new scans for estimating 183 data points within the trajectory. The test was implemented in an indoor basketball court of Miami University in Oxford, Ohio.

For the example shown in Figure 54, a comparison of the trajectories of the ranging measurement system obtained using both the separated real-time localization process and the PPT system is shown in Figure 55.

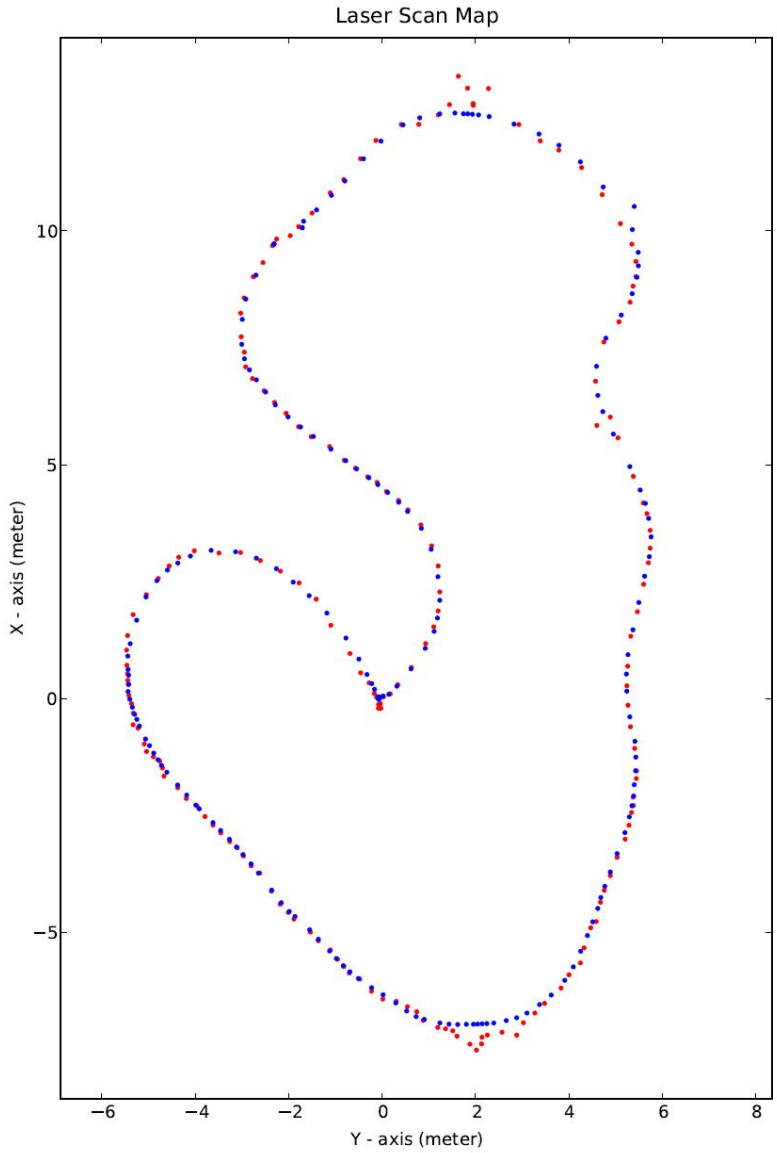


Figure 55. A comparison of the trajectories of ranging measurement system obtained by using both the real-time localization process and the PPT system. The trajectory in red is estimated by implementing the separated real-time localization process, while the trajectory in blue is estimated by using the PPT of the HIVE. The HIVE is located in an indoor basketball court of Miami University in Oxford, Ohio. The RMS error of the trajectory determined by the separated real-time localization process relative to the one determined by the PPT system is approximately 0.13 meters.

The two previous examples were both compared with the measurements obtained by the PPT system, whose results are regarded as the ground truth because of its accuracy. According to the comparison shown in Figure 55, there are occasionally large errors between the trajectories determined by using the separated real-time localization process and the PPT system. These errors are mainly due to mismatching between the map and the newly acquired scans. It is also noted that these errors do not have a marked impact on the accuracy of subsequent estimates, because each process cycle that conducts scan matching is independent from the other process cycles. While scan matching implemented during the separated real-time localization process is successful, there is no marked error when compared with the measurement of the PPT system. On the other hand, when there is mismatching between the map and the newly acquired scan, the error becomes marked and is approximately under 0.75 meters for this particular example. (In this example, the RMS error of the trajectory determined by the separated real-time localization process relative to the one determined by the PPT system is approximately 0.13 meters.)

Unlike the original line-based map-building and localization process, the error produced during a certain cycle of the separated localization process is regarded as a special case for that specific process cycle and is not cumulative.

D. SUMMARY

This chapter presented the improved map-building and localization algorithms obtained by dividing the original algorithm into the uses of two separate processes, the map-building and the localization, for particular types of operating environments. The objectives were to improve the quality of the map being built and to resolve the issue of the accumulated estimation error existing in the original algorithm presented in the previous chapter. Several experiments were conducted to verify the uses of the two separated processes.

The two main contributions of this chapter are summarized as follows.

- A separated map-building process that adopts an offline process approach was constructed to build a map of the operating environment. The map built by using this separated process is more accurate and contains more

details relative to the one built by the original, real-time, line-based map-building and localization process.

- A separated real-time localization process was constructed to estimate the orientation and position of the ranging measurement system by matching the map built beforehand with every newly acquired scan. This separated process is able to resolve the issue of accumulated estimation error that exists in the original, real-time, line-based map-building and localization process. The error produced during a particular cycle of this separated process does not affect the accuracy of subsequent cycles and is not cumulative.

THIS PAGE INTENTIONALLY LEFT BLANK

V. INERTIAL/MAGNETIC SENSOR-BASED LOCOMOTION INTERFACE FOR VIRTUAL ENVIRONMENT SYSTEMS

A. BACKGROUND AND MOTIVATION

A locomotion interface for the virtual environment systems relates the motions made by the user in the real physical environment to the motions in the virtual environment.

The simplest types of locomotion interfaces are the computer mice and keyboards that usually come along with the purchase of computers without the need for support from other peripheral products. When navigating through the virtual environment in first-person[3] types of video games [64], the mouse movements can be related to the changes of the virtual head (or body) orientation, while the key presses can be related to the virtual body movements towards certain directions. A disadvantage of these types of locomotion interfaces is that there is little or no similarity between the real motions and the virtual motions of the users. The users need to practice before getting good control over them.

Joysticks provide users with better control over the motions in the virtual environment. However, they fail to provide a good sense of presence unless the users are driving a virtual vehicle. For the simulated walking motions through the virtual environment, there is still little similarity between the control over the joysticks and over the virtual walking motions.

The locomotion interface of the well-known video game console, Wii [65], released by Nintendo Co, Ltd. provides users with relatively better sense of presence. By utilizing the built-in accelerometers in the Wii remote, it allows users to control the motions of their player characters with their own physical arm gestures.

Another well-known motion sensing device, Kinect for Xbox 360 and Windows PCs [66], [67] released by Microsoft Corporation, provides even better sense of presence. The device utilizes an RGB camera and a depth sensor to provide 3D body motion capture and facial recognition. Different from the Wii, the Kinect has no remote

[3] In video games, first-person refers to a graphical perspective rendered from the player character's point of view [64].

(controller) for the users to hold. The users are able to control the motions of their player characters solely with their own physical body gestures.

One of the disadvantages of the Wii and Kinect-like devices is that the real physical spaces of the users are limited to certain tracking areas of such devices. This may greatly compromise the user's sense of presence. For example, when navigating through the virtual environment by walking, the users can only walk in place to prevent themselves from walking outside of the tracking area, and walking-in-place is still different from natural walking motions.

Some CAVE-like virtual environment infrastructures share the same limit as the Wii or Kinect-like devices. Therefore, in order to allow users to walk naturally through a large-scale virtual environment without significantly changing their actual physical locations, some treadmill designs were proposed for the construction of locomotion interfaces.

One example of the one-dimensional (1D) treadmill designs is the Sarcos Treadport [9], [10], [68] that comprises a large tilting treadmill with an active mechanical tether as shown in Figure 56. The mechanical tether measures the user's physical position and orientation and controls the treadmill speed and the user's location in the virtual environment. Because this type of treadmill works only in one direction, an artificial means is needed for controlling the user's turning motions.

Figure 56. The Sarcos Treadport comprising a large tilting treadmill with an active mechanical tether (from [9]).

The more sophisticated designs that enable users to walk physically in different directions are the 2D treadmill designs. One such design is the Torus Treadmill [11], [12] as shown in Figure 57. The surface of the Torus Treadmill is a combination of 12 treadmills connected side by side. These 12 treadmills can also be driven in a perpendicular direction. Thus, an infinite surface can be generated by moving these treadmill surfaces in both forward-backward and rightward-leftward directions that cancel out the user's movement. While navigating through the virtual environment by walking, the user's real physical location will roughly remain the same at the center of the treadmill surface.

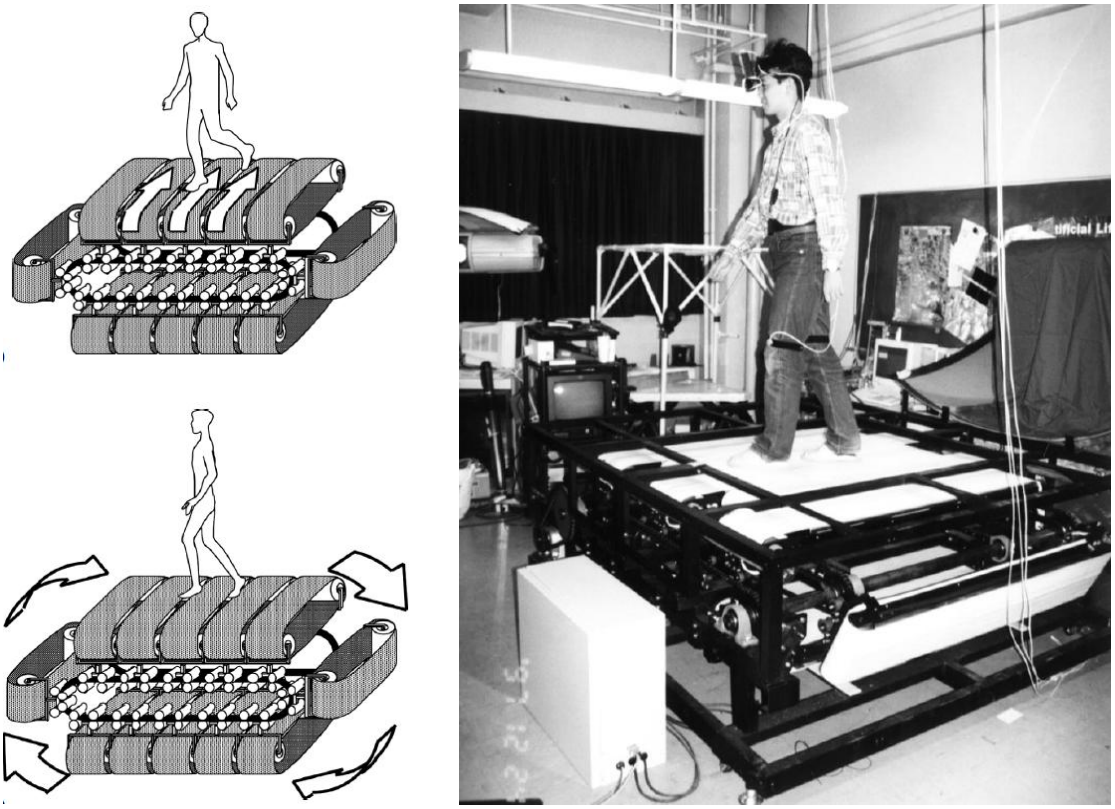


Figure 57. The Torus Treadmill consisting of 12 treadmills connected side-by-side (from [11]).

Another similar design that allows the user to move physically in different directions is the Omni-Directional Treadmill (ODT) [13] shown in Figure 58. It was built for the U.S. Army's Dismounted Infantry Training Program. A study that compared the ODT with a walking-in-place device called LocoX can be found in [69].

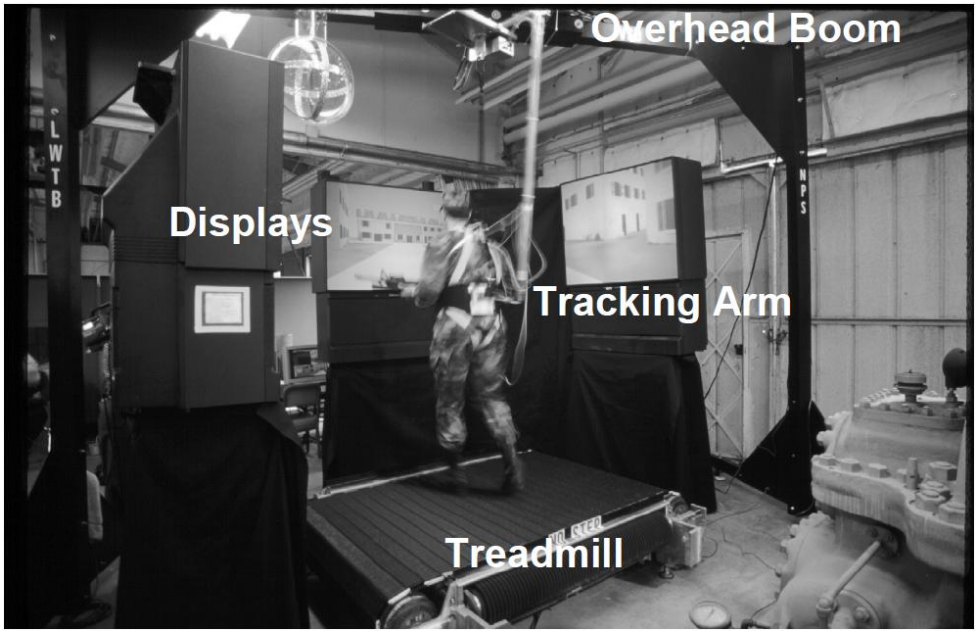


Figure 58. A soldier walks on the Omni-Directional Treadmill (from [13]).

Walking is perhaps the most fundamental example for the user to interact with a virtual environment [70]. The treadmill designs are able to overcome the space limit and to provide the users with a better sense of presence by allowing the users to walk naturally when navigating through the virtual environments. However, locomotion interfaces utilizing the treadmill designs rely heavily on the pre-installed infrastructure in some facilities. The developed virtual environment systems are not portable and not easily accessible to the public.

Motivated by resolving the portability issue, an inertial/magnetic sensor-based locomotion interface will be developed for the construction of the self-contained, portable, and immersive virtual environment systems. Such a locomotion interface will allow the user to navigate through the virtual environments by using natural walking motions. Furthermore, while walking towards a certain direction in the virtual environment, the user will be able to turn his/her head to look around the environment without affecting his/her walking motions.

The proposed locomotion interface consists of two components, the mechanisms for relating the foot (walking) motions and the head motions. The inertial/magnetic

measurement units (IMMUs) are attached to the user's foot and head to measure their motions.

B. MECHANISMS FOR RELATING WALKING MOTIONS

1. Preprocessing of the Sensor Data

The data acquired for the walking motions is from the IMMU attached to the foot. The top view of a notional representation of the IMMU and the foot is shown in Figure 59. In this example, the coordinates of the IMMU follow its own moving body's forward-right-down (FRD) (x-y-z) coordinate system. The subsequent research adopts the same coordinate system for such an IMMU.

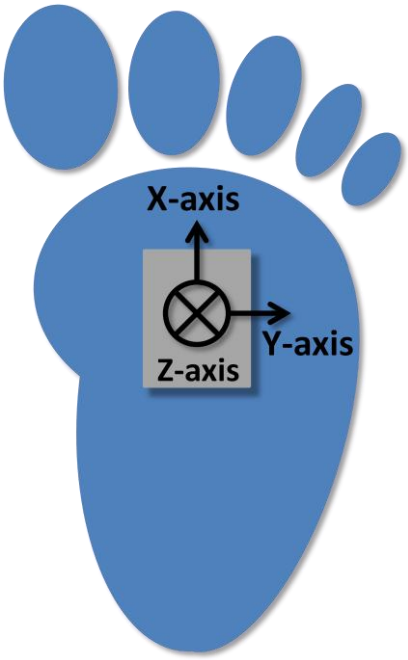


Figure 59. Top view of a notional representation of the IMMU attached to the foot.

The IMMU provides the raw data from its built-in accelerometers, angular rate sensors, and magnetometers along with the orientation represented by the Euler angles (the roll, pitch, and yaw angles) computed by the manufacturer's proprietary algorithm. Note that it is useful to represent the orientation in the form of a quaternion. The use of quaternions enhances the overall efficiency since it avoids the computationally-expensive trigonometric operations performed when the rotation matrices are used instead.

The acceleration measurements are represented in the IMMU's body coordinate system (also called the moving body frame). These measurements need to be transformed and represented in the global (north-east-down) coordinate system (also called the navigation frame) to be used effectively. The orientation represents the attitude of the moving body frame relative to the navigation frame. It serves as the transformation operator between the two coordinate frames.

The orientation (represented by a quaternion) used to transform the IMMU's acceleration measurements is determined using the adaptive gain quaternion-based complementary filter developed in [29], [30]. The orientation obtained by this means is more accurate than the one provided by the IMMU.

The acceleration $\vec{a}^n$ in the navigation frame is determined by transforming the acceleration $\vec{a}^b$ in the moving body frame with the quaternion q. This is accomplished by implementing the quaternion multiplication. Because the component of gravity $\vec{g}$ is contained in the acceleration, its effect needs to be removed after the transformation.

$$\vec{a}^n = q(\vec{a}^b)q^* + \vec{g}, \tag{5.1}$$

where

$$\vec{a}^n = \begin{bmatrix} a_x^n \\ a_y^n \\ a_z^n \end{bmatrix}, \quad \vec{a}^b = \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix}, \quad q = q_0 + \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad q^* = q_0 - \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \text{and} \quad \vec{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}.$$

The 3D vector $\vec{a}^b$ is represented as a pure quaternion with the scalar portion equal to zero during the quaternion multiplication. The result of such a multiplication is also a pure quaternion (i.e., a 3D vector).

As a means of manipulating the 3D object in the virtual environment, the Euler angles (the roll, pitch, and yaw angles) representing the orientation of the object are widely used in most of the 3D graphics application programming interfaces (APIs). Therefore, the quaternion q determined using the adaptive gain quaternion-based complementary filter needs to be converted into the Euler angles. The conversion between the two representations is efficient and straightforward.

2. Gait-Phase Detection

A gait cycle consists of a stance phase and a swing phase. The stance phase of either foot represents the period that this foot is in contact with the ground, while the swing phase represents the period otherwise. A gait phase detection mechanism [28]–[30] is able to distinguish between the stance phase and the swing phase in the gait cycle by comparing the angular velocity of the foot with a specified threshold (1.0 radian/second for the use of this dissertation).

$$|\omega^B| = \sqrt{(\omega_x^B)^2 + \left(\omega_y^B\right)^2}, \tag{5.2}$$

where the term ω^B is the angular velocity of the foot with respect to the moving body frame. The term $|\omega^B|$ is the magnitude of the foot's angular velocity. The angular velocities ω_x^B and ω_y^B are the weights about the x and y axes of the moving body frame. The weight about the z axis is not considered when determining the gait phase. If the magnitude of the foot's angular velocity is greater than the threshold, the foot motion is regarded during the swing phase; otherwise, it is during the stance phase.

The locomotion interface for the walking motions is therefore divided into two portions in accordance with the two separate gait phases. When navigating through the virtual environment, the direction that the user's body is facing is controlled during the stance phase, while the horizontal movement of the user is controlled during the swing phase.

3. Mechanism for Stance Phase

As mentioned earlier, the direction that the user's body is facing in the virtual environment is determined during the stance phase of foot motions. The yaw angle from the orientation data of the foot is used to calculate the virtual body's facing direction. This is based on the fact that the facing direction change of the body about the vertical axis (z axis) is positively correlated with the yaw angle change of the foot while walking.

The orientation data are gathered and processed once while composing each frame[4] of the virtual environment. The virtual body's facing direction D_k for the current frame is equal to its facing direction $D_{k\text{-}1}$ during the previous frame plus the difference $\Delta\alpha$ between the current and previous estimates of the foot's yaw angles.

$$\Delta\alpha = \alpha_k - \alpha_{k\text{-}1}, \tag{5.3}$$

$$D_k = D_{k\text{-}1} + \Delta\alpha, \tag{5.4}$$

where the terms α_k and $\alpha_{k\text{-}1}$ represent the yaw angles of the foot determined during the current and previous frames. The subscripts k and k-1 represent the values determined during the current and previous frames, respectively. Note that the yaw angle of the (physical) foot is relative to the north, while the facing direction of the (virtual) body is relative to its initial direction.

To provide a more consistent scene rotation without sudden jump between two consecutive frames, a threshold is needed to limit the value of yaw angle difference $\Delta\alpha$. The threshold is set in the form of the angular rate. An angular rate of 600 degrees/second is adopted in this dissertation. That is, if the yaw angle difference $\Delta\alpha$ is greater than $(600 \cdot \Delta t)$ degrees or less than $(-600 \cdot \Delta t)$ degrees, it is clamped to $(600 \cdot \Delta t)$ degrees or $(-600 \cdot \Delta t)$ degrees, respectively. The term Δt represents the time difference between the current and previous frames. This may reduce the chance of occurrence of motion sickness to the user.

It is noted that the virtual body's facing direction will not necessarily be equal to the yaw angle of the foot because a relative approach is adopted. This is desired in this research. Such an approach uses the real foot's yaw angle difference to determined how much the virtual body has turned. It will allow further manipulation of the virtual body's facing direction when conducting the obstacle avoidance task, which is investigated later in the next chapter. The human perception will not be able to precisely distinguish the difference between the amounts by which the physical body and the virtual body turned.

[4] The term frame mentioned here has a different meanings from the one used to represent the coordinate system, such as the navigation frame. It represents one of a series of consecutive images rendered by the graphics pipeline.

On the other hand, an absolute approach is to directly set the virtual body's facing direction equal to the real foot's yaw angle. If this approach is used instead, the virtual body's facing direction cannot be further manipulated.

4. Mechanism for Swing Phase

The horizontal movement of the user in the virtual environment is controlled during the swing phase of the foot motions. This is addressed by calculating the horizontal translation of the user during the composition of current frame relative to the user's location and orientation in the previous frame. The transformed IMMU acceleration estimate $\vec{a}^n$ is used for the calculation.

Before the acceleration estimate $\vec{a}^n$ is utilized to determine the translation, a further transformation needs to be conducted. The estimate $\vec{a}^n$ follows the north-east-down coordinate system, while the translation is relative to the user's current orientation and follows the user's forward-right-down coordinate system. A rotation (about the z axis of the north-east-down coordinate system) by the foot's yaw angle α in the negative direction is needed to represent the acceleration in the same coordinate system as the translation. This assumes that the facing direction of the user's physical body coincides with his/her foot's forward direction (x axis).

The transformation operator in terms of a quaternion $q_{\text{-}yaw}$ can be represented as the following:

$$q_{\text{-}yaw} = \cos\left(-\frac{\alpha}{2}\right) + \begin{bmatrix} 0 \\ 0 \\ \sin\left(-\frac{\alpha}{2}\right) \end{bmatrix}. \tag{5.5}$$

The acceleration $\vec{a}^{frd}$ represented in the user's forward-right-down coordinate system can be obtained by transforming the acceleration $\vec{a}^n$ in the north-east-down coordinate system with the quaternion $q_{\text{-}yaw}$.

$$\vec{a}^{frd} = q_{\text{-}yaw}(\vec{a}^n)q^*_{\text{-}yaw}\,, \tag{5.6}$$

where

$$q^{*}_{\text{-}yaw} = \cos\left(-\frac{\alpha}{2}\right) - \begin{bmatrix} 0 \\ 0 \\ \sin\left(-\frac{\alpha}{2}\right) \end{bmatrix}, \quad \text{and} \quad \vec{a}^{\,frd} = \begin{bmatrix} a_x^{frd} \\ a_y^{frd} \\ a_z^{frd} \end{bmatrix}.$$

The horizontal velocities $v_{k,x}^{frd}$ and $v_{k,y}^{frd}$ along the x and y axes of the user's forward-right-down coordinate system during the current frame can be calculated as follows:

$$v_{k,x}^{frd} = v_{k\text{-}1,x}^{frd} + \frac{1}{2}\left(a_{k\text{-}1,x}^{frd} + a_{k,x}^{frd}\right) \cdot \Delta t, \tag{5.7}$$

$$v_{k,y}^{frd} = v_{k\text{-}1,y}^{frd} + \frac{1}{2}\left(a_{k\text{-}1,y}^{frd} + a_{k,y}^{frd}\right) \cdot \Delta t, \tag{5.8}$$

where the subscripts k and k-1 represent the values determined during the current and previous frames, respectively. The term Δt represents the time difference between the current and previous frames.

The horizontal translations x_k^{frd} and y_k^{frd} along the x and y axes of the user's forward-right-down coordinate system during the current frame can also be calculated as follows:

$$x_k^{frd} = v_{k,x}^{frd} \cdot \Delta t, \tag{5.9}$$

$$y_k^{frd} = v_{k,y}^{frd} \cdot \Delta t. \tag{5.10}$$

Due to the effect caused by the drift error in the IMMU's acceleration estimates, the resulting translations calculated using the presented method may oscillate back and forth even if the real movement is towards the same direction. For example, one step forward will always generate positive values for the translation x_k^{frd} during the swing phase by principle. Despite this, negative values may occasionally be produced in practice. Under such a circumstance, the negative values need to be eliminated to provide the user a more consistent sense of movement in the virtual environment. On the other hand, the positive values need to be removed when taking one step back. Therefore, there is a need to determine the desired signs of translations x_k^{frd} and y_k^{frd} (i.e., moving either forward or backward, and either right or left) at the beginning of each swing phase.

The approach used in this dissertation determines the direction of the user's movement by checking the signs of the angular velocities ω_x^b and ω_y^b about the x and y axes of the IMMU's moving body coordinate system at the beginning of each swing phase. For each step forward, a negative value of ω_y^b is expected. A notional representation of the beginning of swing phase is shown in Figure 60. On the other hand, a positive value of ω_y^b means that the user is taking a step backward. A similar approach is used to determine the direction (either towards the right or the left) of the side steps. For example, a positive value of ω_x^b means that the user is taking a side step to the right. Note that a normal step of the walking motions usually consists of the composition of two types of movements (forward-backward and right-left movements) along the x and y axes of the user's forward-right-down coordinate system.

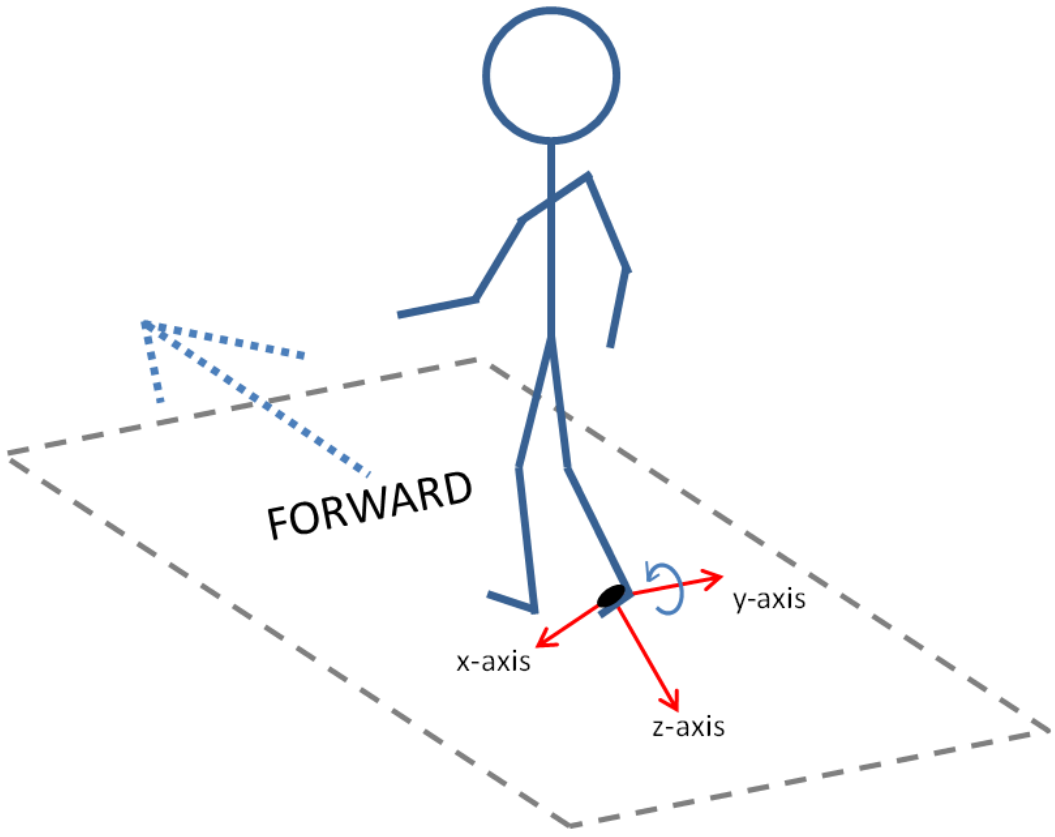


Figure 60. A notional representation of the beginning of swing phase during which a person takes a step forward with the foot mounted with the IMMU. In this example, a negative value is expected for the angular velocity about the y axis of the IMMU's moving body coordinate system.

5. Mechanisms for Walking Motions

The mechanisms for relating the foot (walking) motions during the composition of each frame can be summarized by the flowchart as shown in Figure 61.

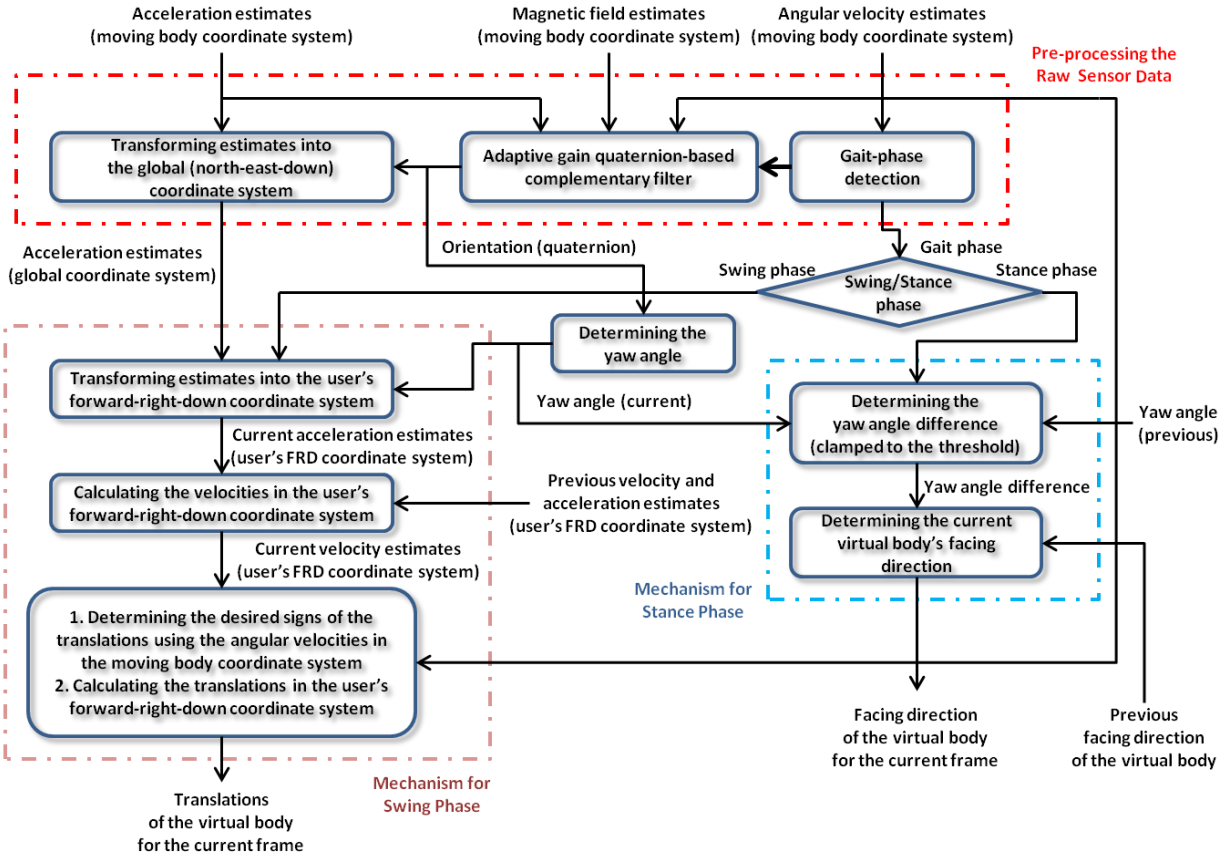


Figure 61. Flowchart of the mechanisms for relating the foot (walking) motions during the composition of each frame.

C. MECHANISMS FOR RELATING THE COMBINATION OF HEAD AND WALKING MOTIONS

The head motions can be incorporated into the mechanisms for relating the walking motions discussed in the previous section. This provides users with a more realistic sense of walking that replicates the real-life walking motions.

Besides the foot motions, the real-life walking motions usually include the head motions that have the following two characteristics:

- The head is able to turn freely (within a certain range not exceeding the limit of human bodies) to look into a different direction other than the body's facing direction and the walking direction.
- Although the head's facing direction and the body's facing direction will not necessarily interfere with each other, the sense of difference between the two directions can be well-perceived by humans. Therefore, the head motions and body motions are not completely independent of each other.

The head motions to be determined are the head's orientation in terms of the Euler angles (the roll, pitch, and yaw angles). The estimates for these angles can be obtained from the IMMU attached to the head-mounted display. The top view of a notional representation of the IMMU attached to the head-mounted display is shown in Figure 62. In this example, the coordinates of the IMMU follow its own moving body's forward-right-down (x-y-z) coordinate system.

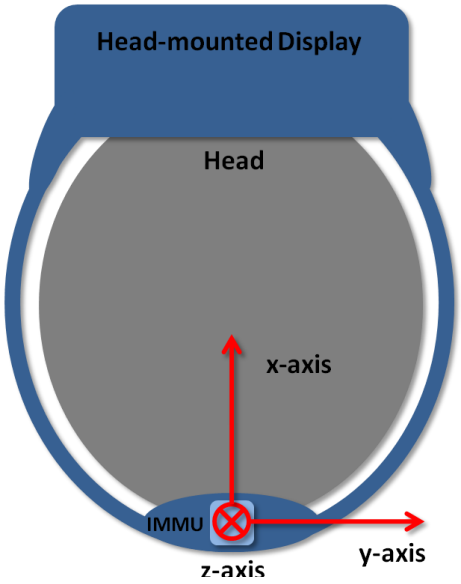


Figure 62. Top view of a notional representation of the IMMU attached to the head-mounted display.

As discussed in the previous section, the orientation for the IMMU attached to the foot is calculated using the adaptive gain quaternion-based complementary filter developed in [29], [30]. This is intended to determine the orientation that is accurate enough to be further used to transform the acceleration estimates. However, for the use of solely determining the head orientation in this section, the orientation (following the

north-east-down coordinate system) provided by the IMMU attached to the head can be directly adopted. For relating the true head motions to the virtual head motions, the orientation accuracy is not crucial as long as the sense provided for the user is realistic.

The virtual head's roll angle for each frame is determined by directly adopting the IMMU's roll angle estimate without further processing. This portion of the head motions is relatively less active in comparison with the changes in the head's yaw and pitch angles.

For determining the virtual head's pitch angle, the direct use of IMMU's pitch angle estimate was originally investigated. However, the virtual environments often oscillated up and down severely due to the walking motions. This would cause great discomfort to the users. To reduce the magnitude of oscillations, a relative approach is used and a threshold value is utilized to limit the virtual head's pitch angle difference $\Delta\beta$ between two consecutive frames.

The virtual head's pitch angle P_k for the current frame is equal to the angle P_{k-1} during the previous frame plus the difference $\Delta\beta$ between the current and previous estimates of the real head's pitch angles.

$$\Delta\beta = \beta_k - \beta_{k\text{-}1}, \tag{5.11}$$

$$P_k = P_{k\text{-}1} + \Delta\beta. \tag{5.12}$$

where the terms β_k and β_{k-1} represent the real head's pitch angles determined during the current and previous frames. The subscripts k and k-1 represent the values determined during the current and previous frames, respectively.

The threshold is set in the form of an angular rate. In this dissertation, an angular rate of 120 degrees/second is adopted. That is, if the pitch angle difference $\Delta\beta$ is greater than $(120 \cdot \Delta t)$ degrees or less than $(\text{-}120 \cdot \Delta t)$ degrees, it is clamped to $(120 \cdot \Delta t)$ degrees or $(\text{-}120 \cdot \Delta t)$ degrees, respectively. The term Δt represents the time difference between the current and previous frames.

It is noted that the virtual head's pitch angle will deviate from the real one by using such an approach. One method to resolve this issue is to make an adjustment per

frame to reduce the discrepancy between two angles. An angular rate of 5 degrees/second is adopted to align the virtual head's pitch angle with the real one gradually.

The relative approach is also used to determine the virtual head's yaw angle. The difference $\Delta\alpha_{real\_head}$ between the current and previous estimates of the real head's yaw angles is obtained as the following:

$$\Delta\alpha_{real\_head} = \alpha_{real\_head,k} - \alpha_{real\_head,k\text{-}1}. \tag{5.13}$$

Again, in order to provide a more consistent scene rotation without sudden jumps between two consecutive frames, a threshold is needed to limit the value of yaw angle difference $\Delta\alpha_{real\_head}$. The threshold is set in the form of an angular rate, which is 600 degrees/second in this dissertation. That is, if the yaw angle difference $\Delta\alpha_{real\_head}$ is greater than $(600 \cdot \Delta t)$ degrees or less than $(-600 \cdot \Delta t)$ degrees, it is clamped to $(600 \cdot \Delta t)$ degrees or $(-600 \cdot \Delta t)$ degrees, respectively.

Assuming that the virtual body's facing directions are the same during the current and previous frames, the virtual head's yaw angle Y_k for the current frame will be equal to the angle $Y_{k\text{-}1}$ during the previous frame plus the real head's yaw angle difference $\Delta\alpha_{real\_head}$. However, the virtual head's yaw angle is always represented relative to the virtual body's facing direction. While the difference of the virtual body's facing direction $\Delta\alpha_{virtual\_body}$ is not necessarily zero, the real head's yaw angle difference $\Delta\alpha_{real\_head}$ is related to the combination of the virtual body's facing direction difference $\Delta\alpha_{virtual\_body}$ and the virtual head's yaw angle difference relative to the virtual body's facing direction $\Delta\alpha_{virtual\_head/virtual\_body}$.

$$\Delta\alpha_{real\_head} = \Delta\alpha_{virtual\_body} + \Delta\alpha_{virtual\_head/virtual\_body}. \tag{5.14}$$

The virtual head's yaw angle Y_k (relative to the virtual body's facing direction) for the current frame can be determined as the following:

$$Y_k = Y_{k\text{-}1} + \Delta\alpha_{virtual\_head/virtual\_body}, \tag{5.15}$$

where

$$\Delta\alpha_{virtual\_head/virtual\_body} = \Delta\alpha_{real\_head} - \Delta\alpha_{virtual\_body}.$$

Based on the experimental investigation, both the virtual body's facing direction and the virtual head's yaw angle are determined using the relative approach, which makes the angles easy to manipulate. The true head motions and body (walking) motions can be related to the motions in the virtual environments in a relatively independent manner. However, as mentioned earlier, the head motions and body motions should not be completely independent. The sense of the virtual head's yaw angle relative to the body needs to be consistent with the real one. Otherwise, some awkward situations may occur. For example, moving forward physically may result in moving backward virtually. Therefore, a mechanism is needed to correct the discrepancy between the real and virtual yaw angles of the head relative to the body.

The correction to such a discrepancy can be conducted during the stance phase of the foot motions. Assuming that the real foot's yaw angle also represents the real body's yaw angle, the real head's yaw angle relative to the body $\alpha_{real\_head/real\_body,k}$ is obtained by calculating the difference between the yaw angle $\alpha_{real\_head,k}$ from the IMMU on the head-mounted display and the yaw angle $\alpha_{real\_body,k}$ from the one on the foot.

$$\alpha_{real\_head/real\_body,k} = \alpha_{real\_head,k} - \alpha_{real\_body,k}\ , \tag{5.16}$$

where the real body's yaw angle $\alpha_{real\_body,k}$ is equal to the real foot's yaw angle $\alpha_{real\_foot,k}$.

The virtual head's yaw angle relative to the body for the previous frame is $Y_{k\text{-}1}$. The difference $\Delta\alpha_{head/body}$ between the real and virtual heads' yaw angles relative to their bodies can be obtained as the following.

$$\Delta\alpha_{head/body} = \alpha_{real\_head/real\_body,k} - Y_{k\text{-}1}. \tag{5.17}$$

As discussed in the previous section, the virtual body's facing direction for the current frame D_k is equal to its facing direction $D_{k\text{-}1}$ during the previous frame plus the difference between the current and previous estimates of the foot's yaw angles.

$$\Delta\alpha_{real\_body} = \alpha_{real\_foot,k} - \alpha_{real\_foot,k\text{-}1}, \tag{5.18}$$

$$D_k = D_{k\text{-}1} + \Delta\alpha_{real\_body}. \quad (5.19)$$

To correct the discrepancy, the modification is applied to adjust the virtual body's facing direction eventually. For this end, the equation (5.18) needs to be modified as the following:

$$\Delta\alpha_{real\_body} = \alpha_{real\_foot,k} - \alpha_{real\_foot,k\text{-}1} - \Delta\alpha_{head/body}. \quad (5.20)$$

By incorporating the mechanism that corrects the discrepancy between the real and virtual yaw angles of the head relative to the body, the real head and body (walking) motions can be replicated for navigating through the virtual environments. The mechanisms for relating the combination of head and foot (walking) motions during the composition of each frame can be summarized by the flowchart shown in Figure 63.

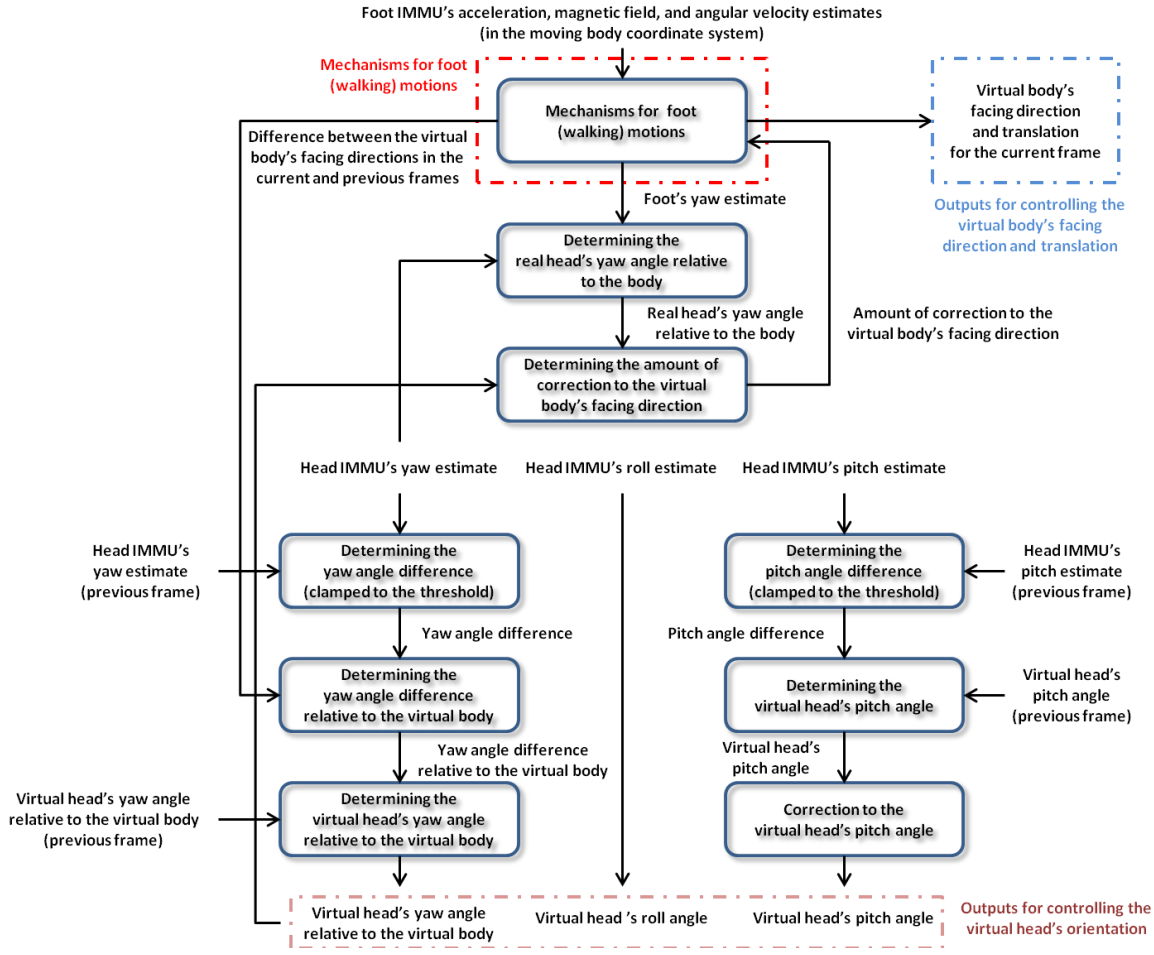


Figure 63. Flowchart of the mechanisms for relating the combination of head and foot (walking) motions during the composition of each frame.

D. SUMMARY

This chapter presented the development of a locomotion interface that relates physical walking motions to virtual walking motions for navigating through virtual environments, disregarding the obstacle avoidance issue in physical environments. The virtual walking motions from the first-person perspective consist of two major sets of motions, the foot (body) motions and the head motions. In correspondence with such motions, the inertial/magnetic measurement units (IMMUs) were attached to both the user's real foot and head for collecting the raw sensing data regarding their motions. Separate mechanisms were constructed for relating such sensing data to the virtual body and head motions. Through the interactions established between the separate mechanisms of the body and head motions, the natural walking motions were able to be replicated in virtual environments.

The main contribution of this chapter is providing an approach to developing an inertial/magnetic sensor-based locomotion interface that allows the user to use natural walking motions to navigate through virtual environments. The locomotion interface developed contributes to the construction of self-contained, portable, and immersive virtual environment systems. The method used to address the obstacle avoidance issue in a physical environment with a limited size is discussed in the next chapter.

VI. INERTIAL/MAGNETIC SENSOR-BASED LOCOMOTION INTERFACE INCORPORATED WITH A RANGING MEASUREMENT SYSTEM

A. BACKGROUND AND MOTIVATION

In the previous chapter, an inertial/magnetic sensor-based locomotion interface for virtual environment systems was developed. Such a locomotion interface allows the user to walk naturally to navigate through virtual environments. It is noted that the obstacle avoidance problem in physical environments has not yet been considered during the interface development. This may not be an issue when operating in a physical environment with dimensions larger than the dimensions of the virtual environment. However, one of the key purposes of utilizing virtual environments is often to overcome the limitation in the dimensions of physical environments. It is necessary to always consider the virtual environments larger than the physical space available when developing the locomotion interface. A mechanism for the user to avoid obstacles in the physical environment therefore needs to be incorporated into the mechanisms of the locomotion interface.

Obstacles in this dissertation are referred to as static (or moving) objects that may block the user's course of walking motions. Common examples may include the boundaries (walls) and the furniture (tables or chairs) in the physical environment.

As briefly discussed in Chapter II (Section D), redirected walking [42]–[44] may be adapted to avoid obstacles. It was designed to rotate the entire virtual environment around the user's viewpoint imperceptibly to steer the user's walking direction away from obstacles. The injected rotation rate can be made to be proportional to the magnitude of user's virtual linear and angular velocities, which can be determined by utilizing the inertial/magnetic sensor-based system (locomotion interface) developed in the previous chapter. For example, a higher rotation rate can be injected when the magnitude of the linear or angular velocity is higher.

The original redirected-walking mechanism [42]–[44] was created under the assumption that the user's physical environment is known and fixed. For example, a

CAVE-like environment or a space equipped with a pre-installed optical sensor-based measurement system was utilized to contain the user in some kind of tracking area. The physical position and orientation of the user in such a tracking area can be obtained relatively directly for conducting obstacle avoidance. However, for the system configuration of the locomotion interface aiming at portability in this study, the operating physical environment is an arbitrary open space rather than a fixed tracking area with the pre-installed infrastructure. In order to utilize the concept of redirected walking for obstacle avoidance, one of the key tasks of the locomotion interface is to collect and process the information regarding the physical environment surrounding the user. Since a ranging measurement system has been proven feasible for collecting such information, as discussed in Chapters III and IV, it can be incorporated as a substitute for the various infrastructures that offer a complete knowledge of the physical environment for conducting redirected walking.

There are two different approaches to solving the obstacle avoidance problem, the absolute approach and the relative approach. The absolute approach requires a complete knowledge of the user's physical environment (such as the whereabouts of obstacles) and the user's position and orientation in this environment. On the other hand, the relative approach only requires the user's local information directly measured by the ranging measurement system, such as the distances and directions of obstacles relative to the user. The original redirected-walking mechanism falls into the category of the absolute approach.

Motivated to resolve the obstacle avoidance issue, in this chapter an improved redirected-walking mechanism is developed to utilize the environment information collected by a ranging measurement system to operate in accordance with the mechanisms of the inertial/magnetic sensor-based locomotion interface proposed in the previous chapter. Offering better processing efficiency for real-time implementation, the mechanism will be based on the relative approach, which utilizes only the local information from one single scan of the ranging measurement system at the current moment.

B. REDIRECTED WALKING FOR OBSTACLE AVOIDANCE

1. Original Redirected-Walking Mechanism

The redirected-walking mechanism is a technique that injects an angular velocity into virtual environments to steer the user's walking direction gradually towards a desired target location in the physical environment. The desired result of the original redirected-walking mechanism can be described by Figure 64. Assuming that the user initially faces the wall in the physical environment, as the user walks closer and closer to the wall, the virtual trajectory may eventually go through the wall, while the real trajectory will be gradually steered away from the wall and towards a specified target location.

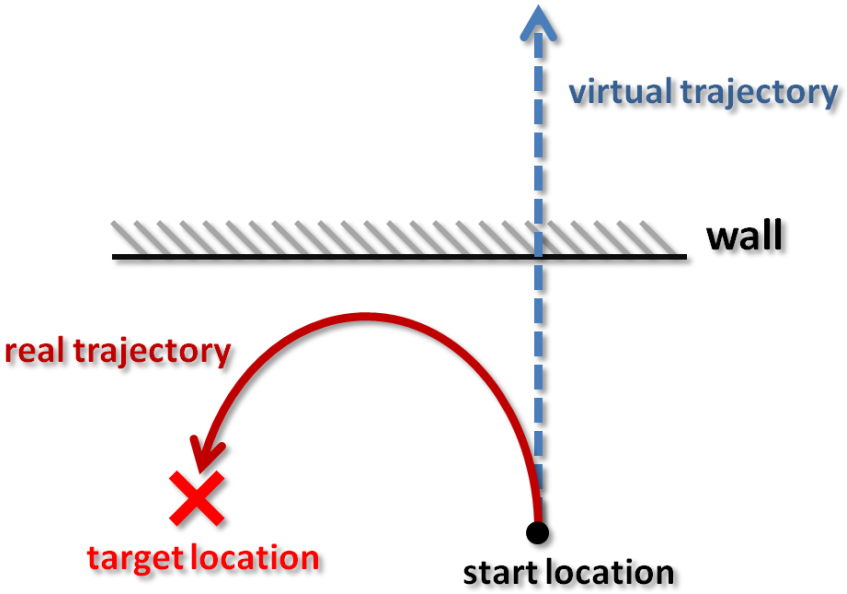


Figure 64. Desired result of the original redirected-walking mechanism. As the user walks closer and closer to the wall in the physical environment, the virtual trajectory may eventually go through the wall, while the real trajectory will be gradually steered away from the wall and towards a specified target location. It is noted that the virtual and real trajectories initially overlapped with each other.

According to [42] and [44], the original redirected-walking mechanism uses the magnitude of user's linear and angular velocities as input parameters to determine the candidates of rotation rates to be injected to the rotation of virtual environment. A small baseline rotation rate is injected even when the user is standing still. The maximum of the

three candidates of rotation rates is chosen and scaled by a direction coefficient, which is determined by computing the sine of the angle between the user's current heading and the target location in the real physical environment. The scaled value is then compared with a certain threshold value that is the angular velocity which seemed imperceptible to the user. If it does not exceed the threshold, it is used as the angular velocity to be injected into the rotation of the virtual environment. Otherwise, it is clamped to the threshold value. The block diagram for such a mechanism is shown in Figure 65.

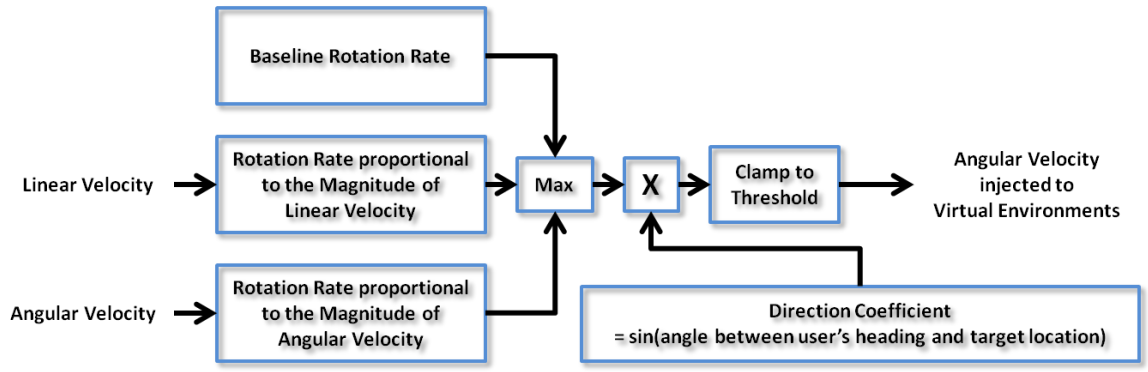


Figure 65. Block diagram of the original redirected-walking mechanism (adapted and revised from [42] and [44]).

One of the issues with this mechanism is that when the user is heading directly away from the target location (often the center of real physical environment), the system does not steer him/her back toward it. The reason is that the angle between the user's heading and the target location is now 180 degrees, which causes the direction coefficient equal to zero. Therefore, an approach introduced in [44] is to pre-allocate multiple target locations. When the user is heading directly away from one target location, the system will automatically select another that is more convenient for conducting the redirected-walking mechanism.

However, specifying a single or multiple fixed target locations is not always optimal for conducting redirected walking. A better concept is to continuously steer the user towards a direction that offers more vacant space instead of some particular locations to better utilize the space available and to reduce the unnecessary rotation rate

being injected. Before discussing the realization of such a concept, the form of the environment information being collected and provided by the ranging measurement system is briefly reiterated.

2. Information Provided by the Ranging Measurement System

The data collected by the ranging measurement system typically consist of an array of range measurements. Since the direction of each measurement can be determined based on the configuration of sensor units, these measurements can be represented as a set of vectors relative to the center of the ranging measurement system in the two-dimensional or three-dimensional coordinate system (either in the polar or Cartesian form).

As shown in Figure 66, each range measurement can be represented as a vector $\vec{r}$ with the length d and the bearing θ in the two-dimensional polar form. The vector $\vec{r}$ can also be represented in the two-dimensional Cartesian form.

$$\vec{r} = \begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} d \cdot \cos\theta \\ d \cdot \sin\theta \end{bmatrix}. \tag{6.1}$$

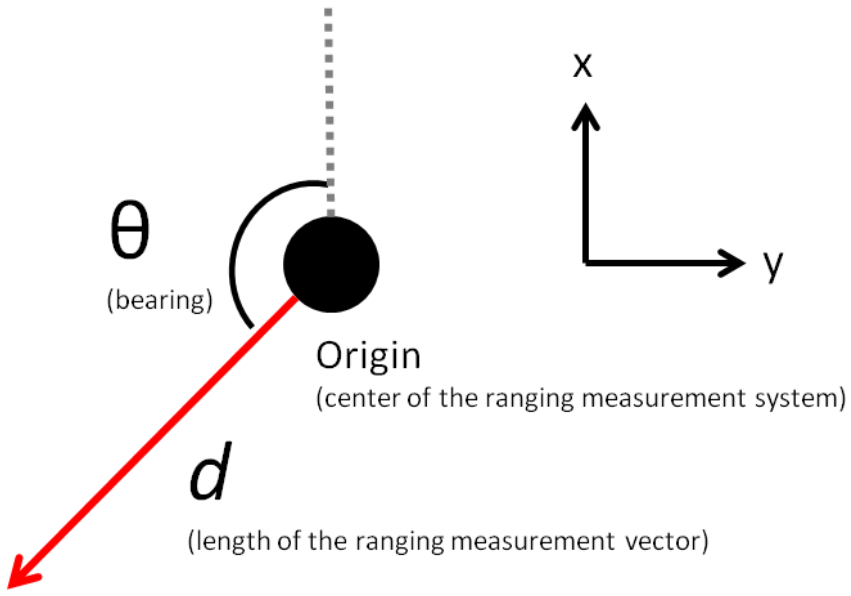


Figure 66. Notional representation of the range measurement as a vector with the length d and the bearing θ.

3. Dynamically-Updated Target Direction Substituted for Particular Predefined Locations in the Redirected-Walking Mechanism

The intended use of the redirected-walking mechanism is to steer the user away from obstacles (such as walls) in the physical environment. The original redirected-walking mechanism specifies a target location (usually the center of the tracking area) or a set of target locations by turns. Ideally, the user will be redirected to walk towards such a location eventually.

However, specifying a single or multiple fixed target locations is not always optimal for conducting redirected walking as discussed earlier. A better approach is to continuously steer the user towards a direction that offers more vacant space instead of some particular locations. Such a dynamically-updated target direction can be determined by constructing the potential field [71] of the measurement vectors provided by the ranging measurement system.

For example, a notional representation of eight range measurements at a given moment is illustrated as shown in Figure 67. The terms R1 through R8 represent the distances between the ranging measurement system (carried by the user) and the obstacles. Note that the number of range measurements will be based on the configuration of the ranging measurement system. The use of eight measurements in this example is only intended for illustrating the concept.

The repulsive potential field of such range measurements can be illustrated as shown in Figure 68. It consists of all the repulsive forces represented by the reciprocals of the range measurements. The combined force of the potential field is shown in Figure 69. The direction of the combined force can be utilized as the target direction in the redirected-walking mechanism. The magnitude is irrelevant for the use of the mechanism.

Since the magnitude of the combined force is irrelevant, the combined force utilized in this chapter is determined by directly calculating the sum of the vectors representing all of the range measurements. The resulting target direction is similar to the one obtained by calculating the combined force of the repulsive potential field. Assuming there are n range measurements, the combined force $\vec{F}$ is calculated as the following:

$$\vec{F} = \sum_{k=1}^{n} \vec{r}_k. \tag{6.2}$$

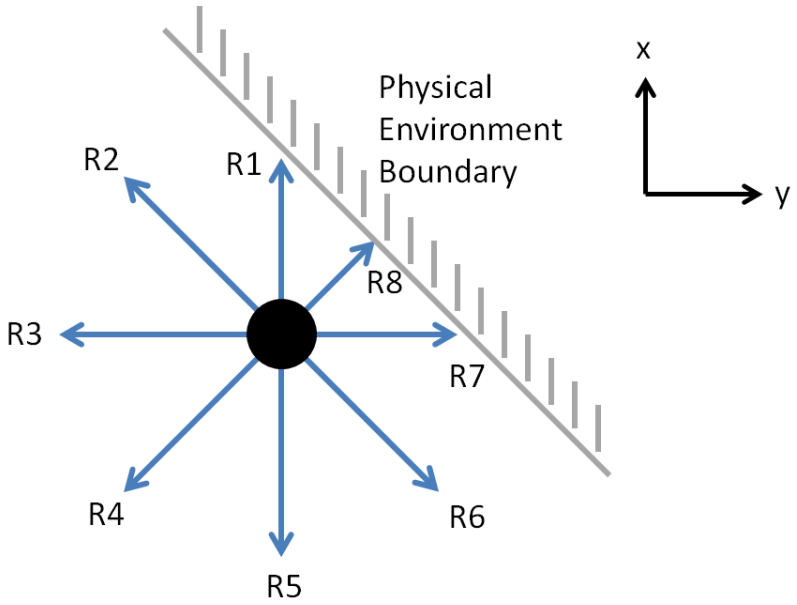


Figure 67. Notional representation of eight range measurements around the ranging measurement system. The terms R1 through R8 represent the distances between the ranging measurement system and the obstacles. If there is no obstacle within the maximum sensing range of the system, the max range value will be represented. The filled circle represents the user carrying the ranging measurement system. Note that the number of range measurements will be based on the configuration of the ranging measurement system. The use of eight measurements in this example is only intended for illustrating the concept.

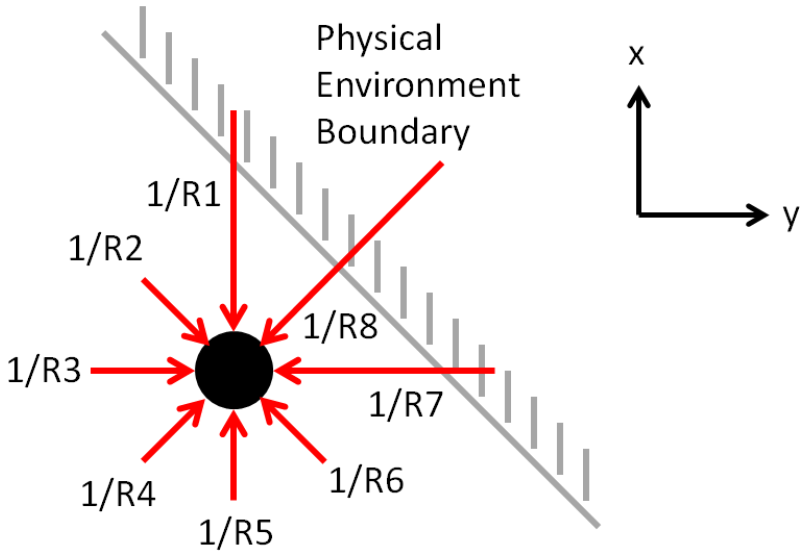


Figure 68. Repulsive potential field created by the eight notional range measurements around the ranging measurement system.

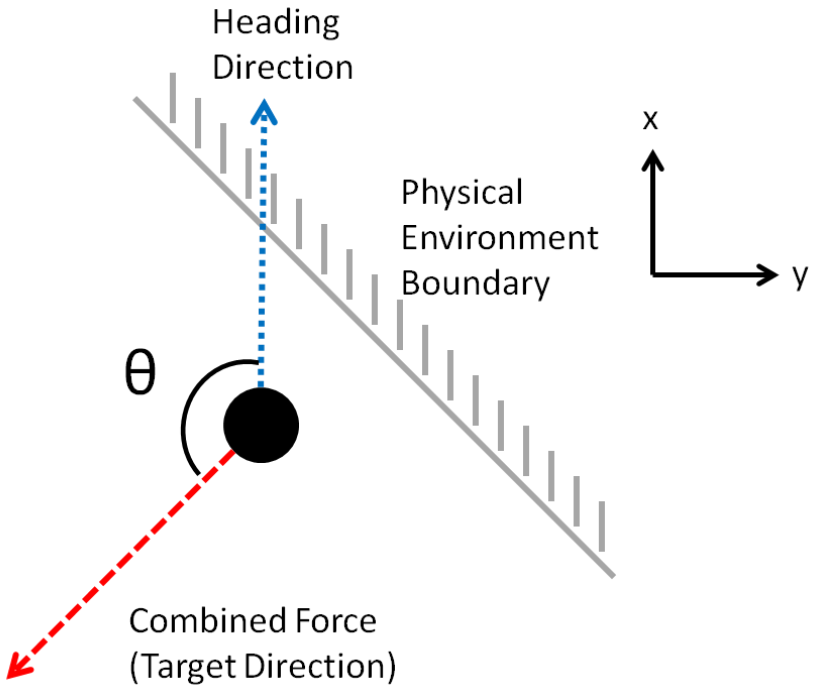


Figure 69. Combined force of the potential field. The direction of the combined force can be utilized as the target direction in the redirected-walking mechanism. The term θ is the angle between the user's heading direction and the target direction in the physical environment.

4. Potential Field-Based Redirected-Walking Mechanism

The potential field-based redirected-walking mechanism can be constructed by utilizing the target direction determined by calculating the combined force of the potential field.

The block diagram of the mechanism is shown in Figure 70. The term θ is the angle between the user's heading direction and the target direction, the same as illustrated in Figure 69. Different from the original redirected-walking mechanism, the direction coefficient in the potential field-based mechanism is modified and replaced with the sine of half the angle θ. Since the angle θ falls between ± 180 degrees, the use of the sine of the half angle is intended to render the value of direction coefficient between ± 1. In this way, the more the user's heading direction is turned away from the target direction, the

greater the direction coefficient will be produced. For example, the max angle (±180 degrees) will cause the maximum direction coefficient (±1) to be produced in the potential field-based mechanism, instead of 0 in the original mechanism. The angle θ is calculated as the following:

$$\theta = \tan^{-1}\left(F_y / F_x\right), \tag{6.3}$$

where the terms F_x and F_y are the weights of the combined force $\vec{F}$ in the x and y axes, respectively.

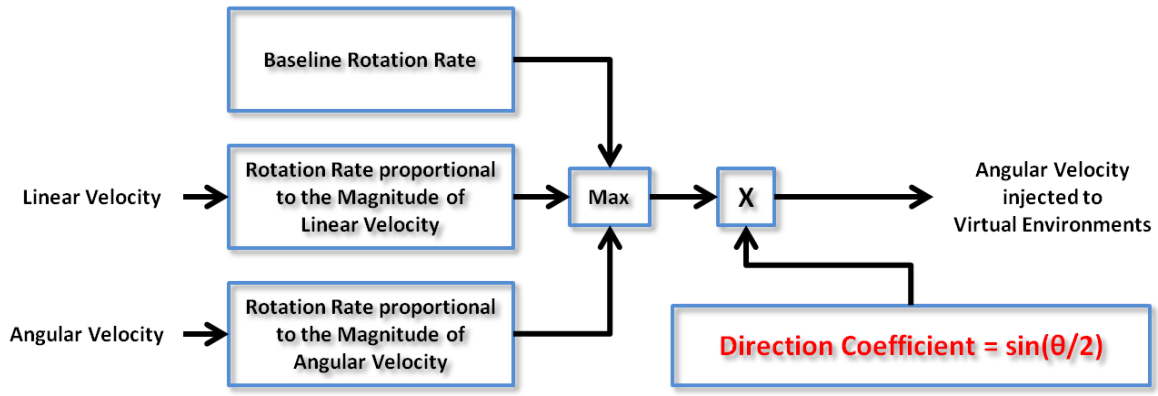


Figure 70. The potential field-based redirected-walking mechanism. The direction coefficient is replaced with the sine of the half angle between the user's current heading direction and the target direction.

The scaled value (calculated by multiplying the maximum of the three candidates of rotation rates with the direction coefficient) is not further compared with a threshold value in the potential field-based mechanism, while such a procedure was conducted in the original mechanism. Such an extra comparison step that clamps the scaled value to the threshold does not offer a noted effect to the redirected-walking result and is therefore omitted.

A rotation rate of 1.0 degree/second is adopted as the baseline rotation rate. Such a rotation rate was experimentally determined in [44] as the detection threshold while the user is standing still. The factors for determining the injected rotation rates proportional to the magnitudes of linear and angular velocities can be adjusted for different operating

environments. In this study, the values of 13.0 and 0.2 are adopted. The two rotation rates can be calculated as follows:

$$\omega_{proportional\ to\ mag(linear\ velocity)} = 13.0 \cdot mag(linear\ velocity), \quad (6.4)$$

$$\omega_{proportional\ to\ mag(angular\ velocity)} = 0.2 \cdot mag(angular\ velocity). \quad (6.5)$$

After the angular velocity to be injected is determined for the current frame, the system (i.e., the locomotion interface) will induce a virtual environment rotation during such a frame. The relationship between the system-induced virtual environment rotation and the user-adjusted walking direction change is illustrated in Figure 71. (It should be noted that the part describing the user's decision/action is only one particular case used for explaining the concept. Since the human decision/action is often unpredictable, this may not be true for describing all cases.) The system part is a per-frame operation. As the amount of the accumulated virtual environment rotation induced by the system becomes larger, the user will notice that his/her walking direction deviates from the original route. Ideally, the user will perceive such a deviation as his/her own motion. In order to follow his/her original route, the user will adjust his/her walking direction. This is the way used by the redirected-walking mechanism to steer the user's walking direction.

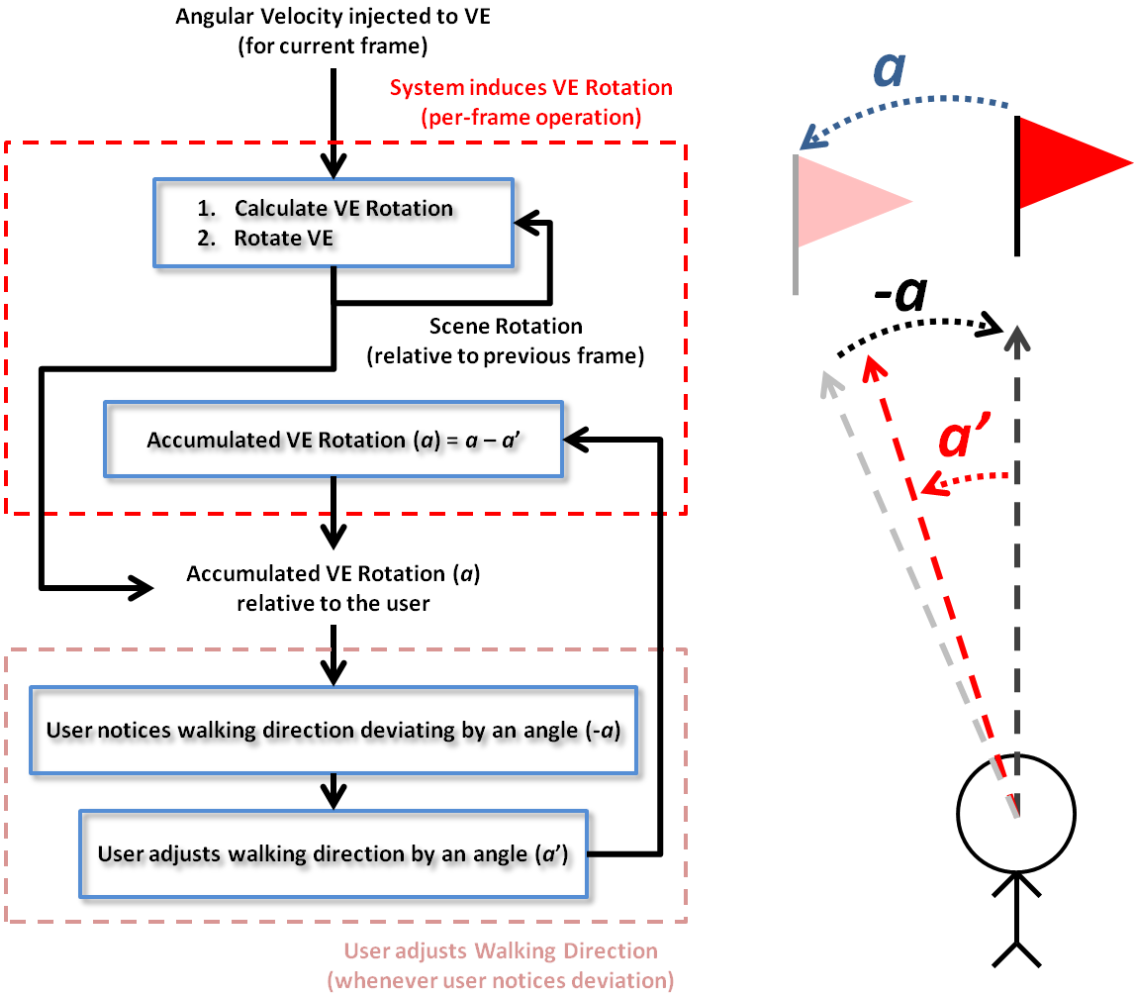


Figure 71. Relationship between the system-induced VE rotation and the user-adjusted walking direction change.

5. Mechanisms Addressing Particular Conditions Incorporated in the Potential Field-Based Redirected-Walking Mechanism

As discussed earlier, the target direction is determined by calculating the combined force of the potential field formed by the range measurements. Ideally, the redirected-walking mechanism will steer the user towards the target direction to prevent the user from colliding with the obstacles, such as walls. However, the user may occasionally walk too close to the obstacles. Under such a circumstance, extra conditions for determining the target direction need to be included in the mechanism. This can be achieved by monitoring the values of the range measurements. The following two

conditions are used to decide whether or not the calculated target direction needs to be replaced and what the replacement will be.

- In coping with the short distance (from obstacles) on the user's left or right side, if the value of the minimum range measurement is less than a threshold value, the new target direction is directly opposite to that of the vector representing such a range measurement. This is intended to accelerate the process of redirecting the user's physical walking direction away from obstacles. Note that the threshold value can be adjusted for different operating environments. A value of 4.0 (in meters) is adopted in this study.
- In coping with the short distance (from obstacles) in front of the user, if the range measurement of the front step is shorter than a range threshold, and the magnitude of the angle between the original target direction and the user's current heading direction is smaller than the angle threshold, the new target direction adopted is directly opposite to the user's current heading direction. This renders the value of direction coefficient either $+1$ or -1 according to the sign of the angle between the original target direction and the user's heading direction. Such a value will be in effect until the range measurement of the front step becomes longer than the range threshold again. This situation usually occurs when the user is walking directly towards a wall with the potential field of the range measurements being approximately symmetric on both the user's sides, which causes the direction of combined force (i.e., target direction) to always point forward. The means adopted is to keep steering the user's walking direction to the right (direction coefficient = $+1$) or to the left (direction coefficient = -1) until the user's forward space becomes open again. Note that the values of the range and angle thresholds can be adjusted for different operating environments. In this study, a value of 10.0 (in meters) is used for the range threshold, while a value of 5.0 (in degrees) is used for the angle threshold.

The flowchart of the process cycle including the two conditions for determining the target direction is illustrated in Figure 72.

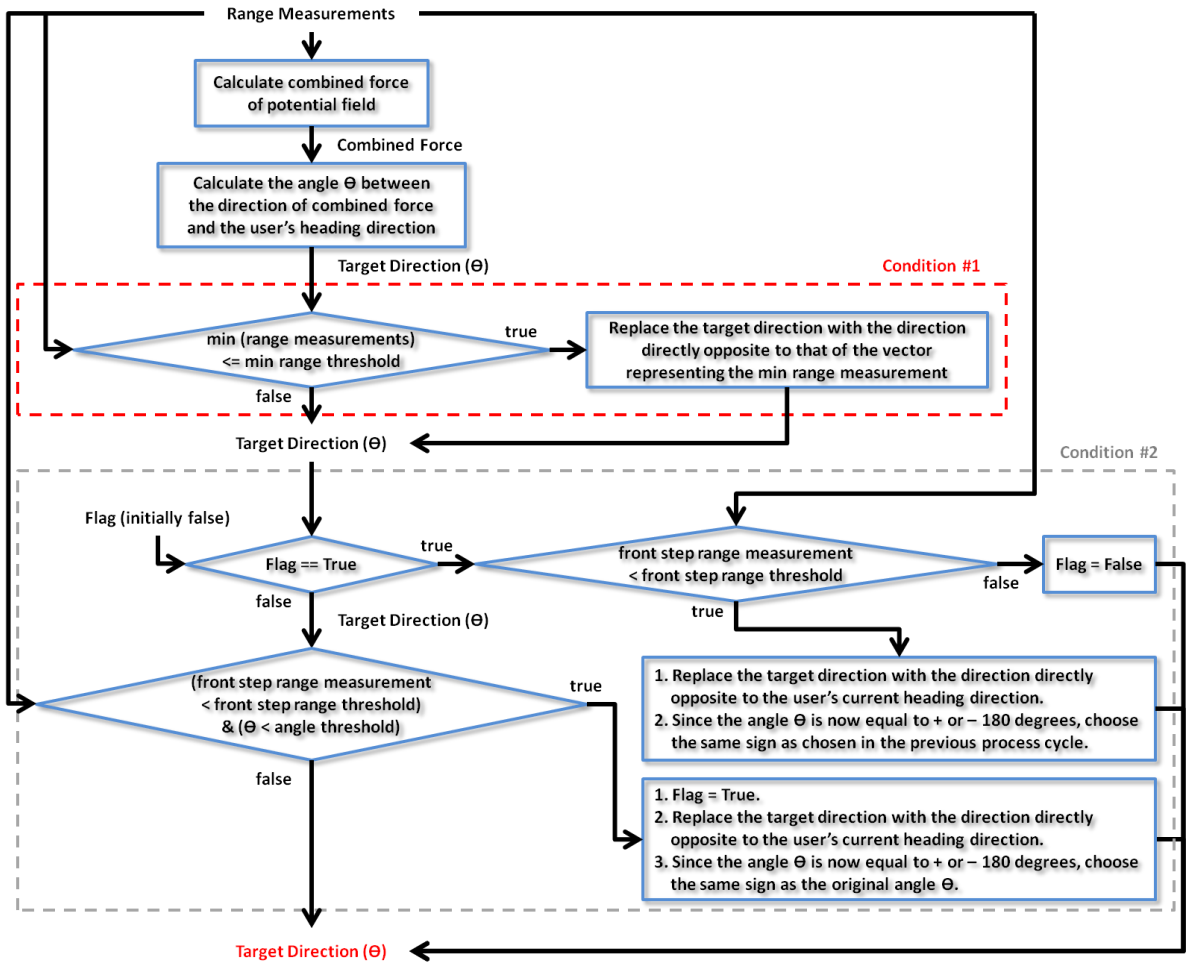


Figure 72. Flowchart of the process cycle determining the target direction.

C. EXPERIMENT RESULT

The result of an experiment using the locomotion interface developed in the previous chapter incorporated with the potential field-based redirected-walking mechanism is shown in Figure 73. The experiment was conducted in the Barbara McNitt Ballroom of the Naval Postgraduate School, as shown in Figure 74. In this experiment, the user wearing the head-mounted display was immersed in a virtual environment of a farm, and was instructed to walk in a straight line path.

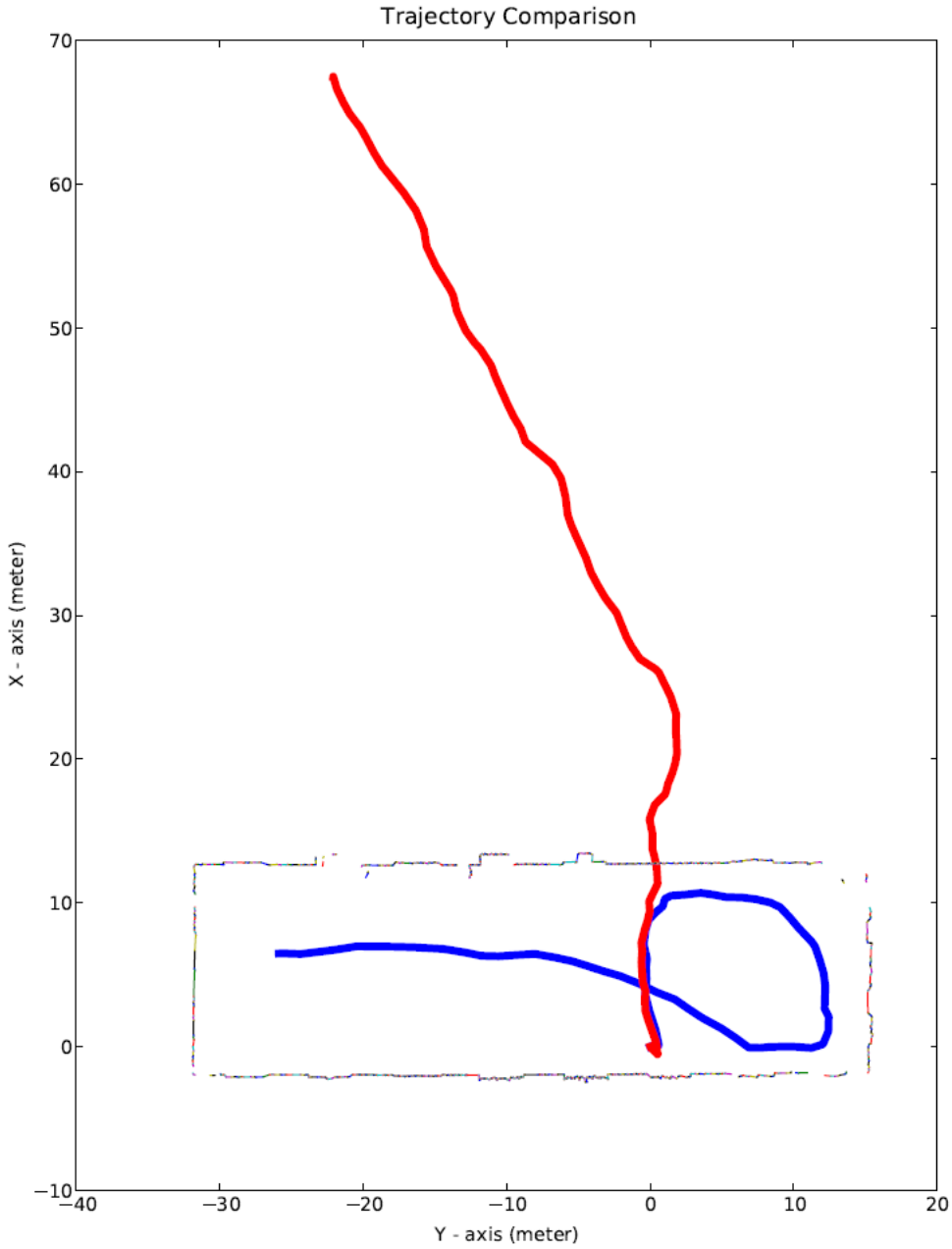


Figure 73. Comparison between the real trajectory (in blue) and the virtual trajectory (in red) of the user. Both of the trajectories start at (0, 0). The line-based map represents the boundary of the physical environment (the Ballroom of the Naval Postgraduate School). The approximate dimensions of the physical environment are 48 x 15 meters.

The line-based map represents the boundary of the physical environment where the experiment was conducted. It was obtained beforehand by conducting the map-building process discussed in Chapter IV. The curve in red represents the virtual

trajectory that was recorded using the built-in methods of the Panda3D and Python libraries. The curve in blue represents the real trajectory in the physical environment. It was determined by conducting scan matching between the map built beforehand and every scan taken throughout the experiment by the ranging measurement system. Since the potential field-based redirected-walking mechanism adopts a relative approach, the localization process is not required during operation. Therefore, the real trajectory was built off-line only for visualizing the redirected-walking result. It is noted that the real trajectory and virtual trajectory coincide at $(0, 0)$ at the beginning of the experiment. As the experiment progresses, while the virtual trajectory follows a more or less straight path extending beyond the physical boundary, the real walking direction was constantly steered away from the boundary of the physical environment regardless the virtual walking direction. The locomotion interface combined with the potential field-based redirected-walking mechanism is thus verified for the use of virtual environment systems to overcome the limitation in the dimensions of the physical environments.

Figure 74. The physical environment for conducting the experiments using the locomotion interface developed in the previous chapter incorporated with the potential field-based redirected-walking mechanism. The picture was taken in the Barbara McNitt Ballroom of the Naval Postgraduate School.

The result of another experiment conducted in the same physical environment is shown in Figure 75. In this experiment, the user was instructed to walk randomly in the virtual environment. The starting position for both the real and virtual trajectories is at (0, 0). It is seen that the real trajectory shown in blue is confined within the physical boundary as it should be. The virtual trajectory shown in red extends far beyond the boundary, which indicates that the user is allowed to explore a virtual environment much larger than the physical environment.

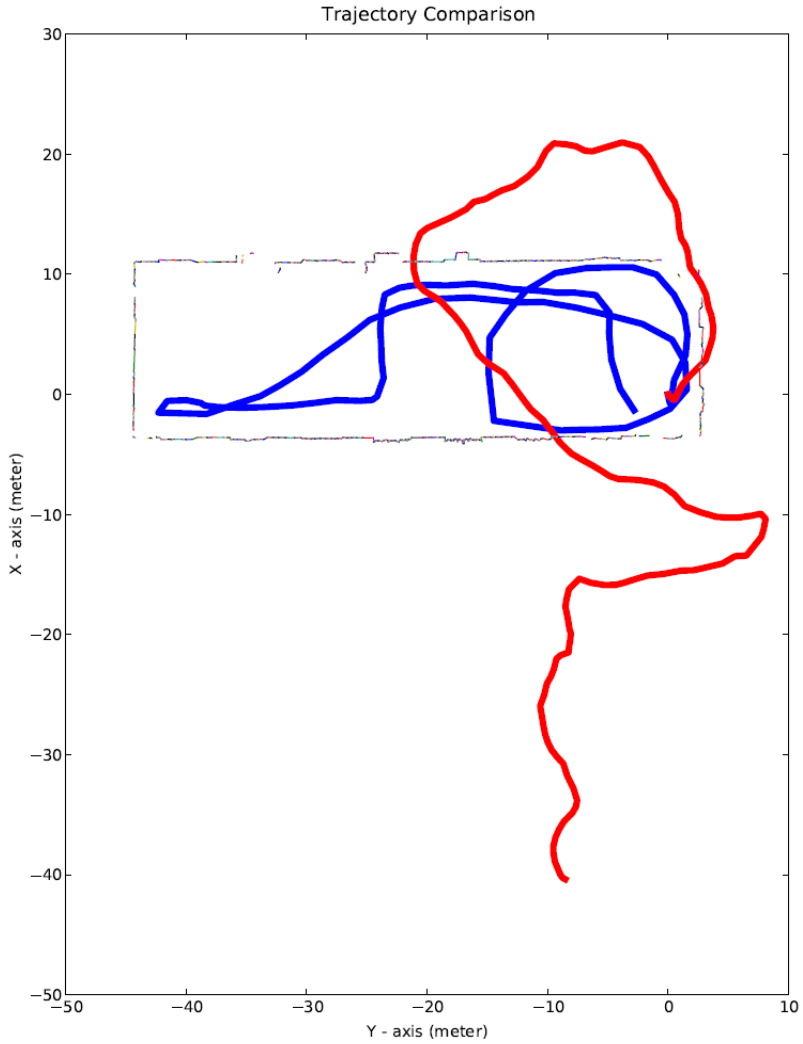


Figure 75. Comparison between the real trajectory (in blue) and the virtual trajectory (in red) of the user. Both of the trajectories start at (0, 0). The line-based map represents the boundary of the physical environment (the Ballroom of the Naval Postgraduate School). The approximate dimensions of the physical environment are 48 x 15 (meters).

To further demonstrate the utility of the locomotion interface and the redirected-walking mechanism, another set of experiments was conducted in a different physical environment, that is, in the basketball court located in the Fitness Center Building of the Naval Postgraduate School, as shown in Figure 76. The real and virtual trajectories from two of the experiments conducted in this environment are presented in Figure 77 and Figure 78. The results were similar to those in the previous two experiments conducted in the Ballroom.

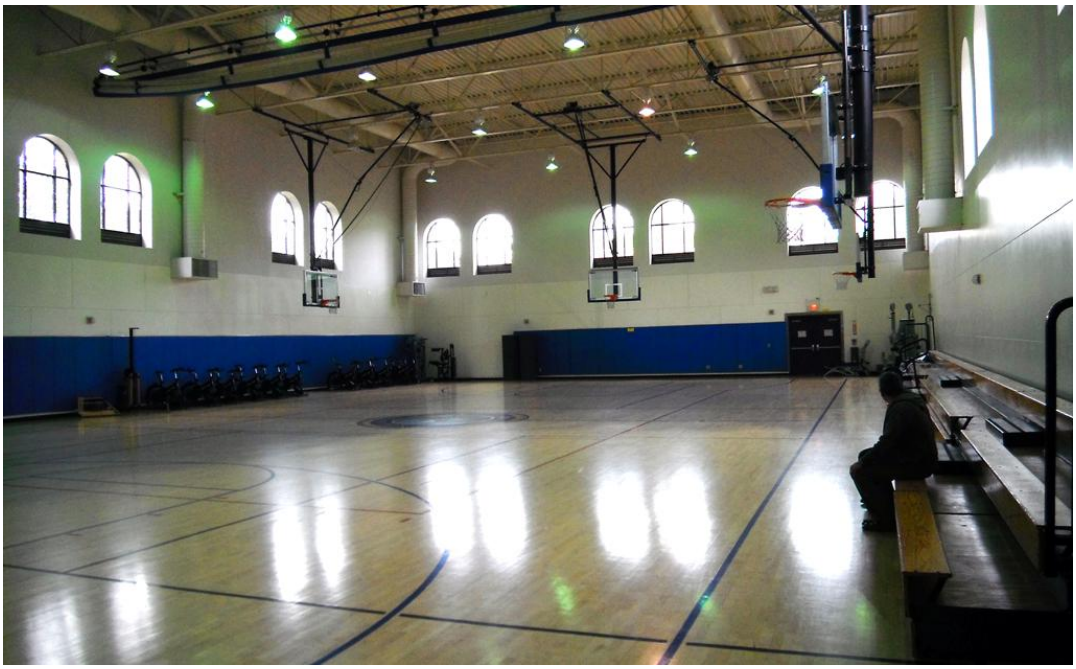


Figure 76. The physical environment for conducting the experiments using the locomotion interface developed in the previous chapter incorporated with the potential field-based redirected-walking mechanism. The picture was taken in the basketball court located in the Fitness Center Building of the Naval Postgraduate School.

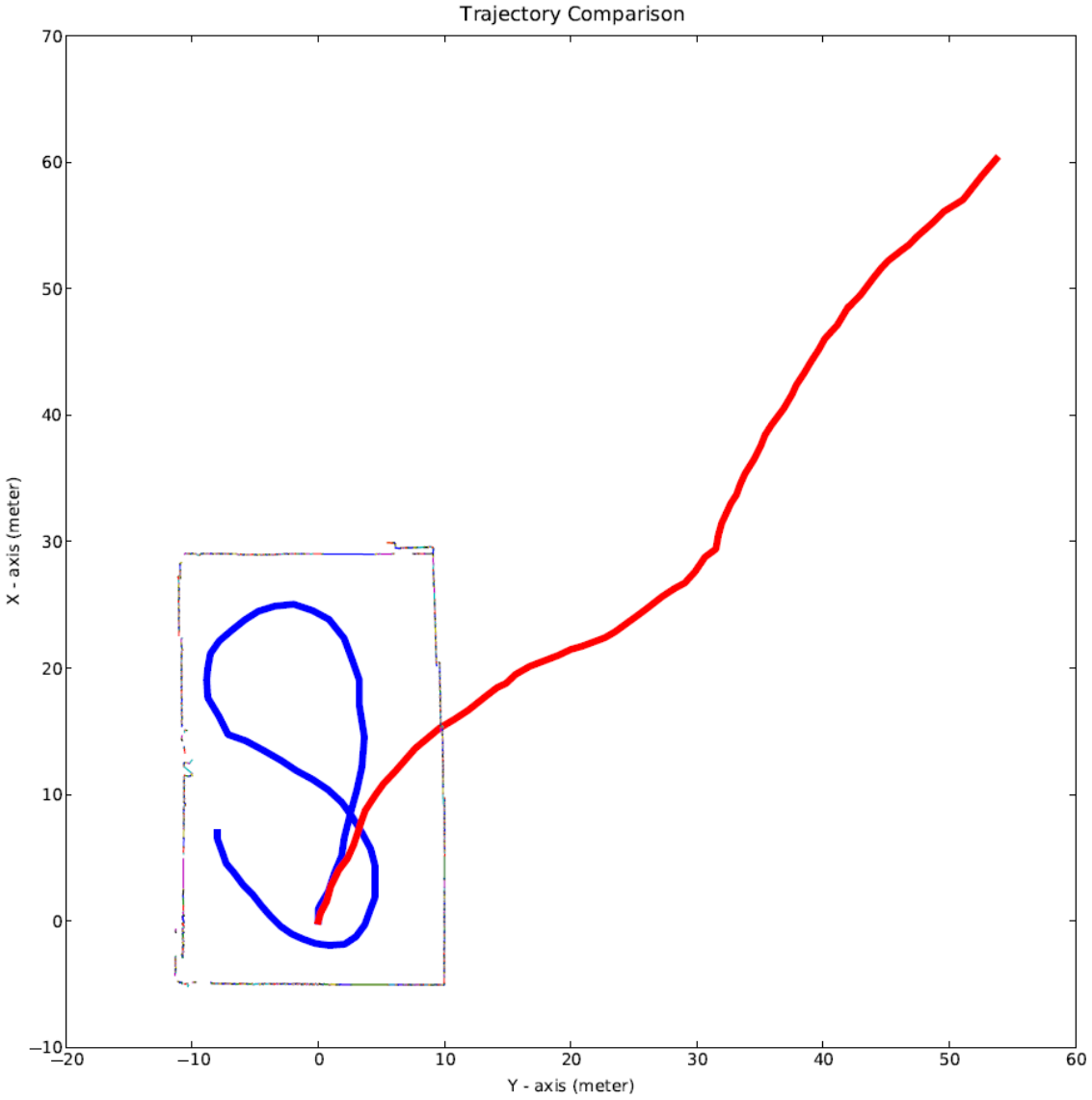


Figure 77. Comparison between the real trajectory (in blue) and the virtual trajectory (in red) of the user. Both of the trajectories start at (0, 0). The line-based map represents the boundary of the physical environment (the basketball court in the gym of the Naval Postgraduate School). The approximate dimensions of the physical environment are 22 x 35 (meters).

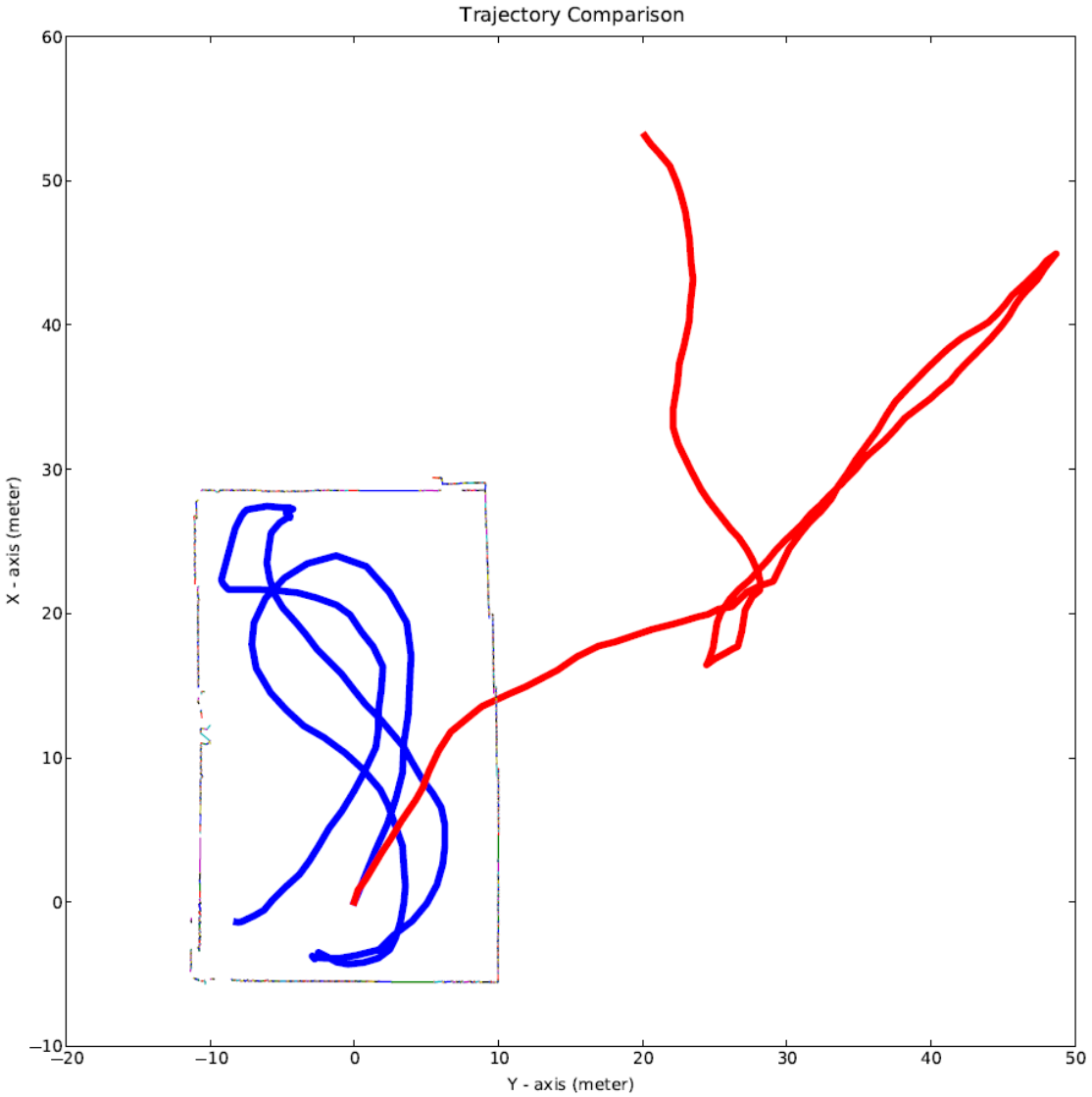


Figure 78. Comparison between the real trajectory (in blue) and the virtual trajectory (in red) of the user. Both of the trajectories start at (0, 0). The line-based map represents the boundary of the physical environment (the basketball court in the gym of the Naval Postgraduate School). The approximate dimensions of the physical environment are 22 x 35 (meters).

D. LIMITATIONS

The inertial/magnetic sensor-based locomotion interface integrated with the potential field-based redirected-walking mechanism can be used to overcome the space limitation of the physical environment. The user is able to navigate through virtual environments that are much larger than the available physical space. For safety

considerations, such an integrated locomotion interface needs to be operated under the following conditions and limitations:

- The use of the integrated locomotion interface requires a flat level physical space. Since the user using a head-mounted display is not able to see the physical environment, any uneven surface may cause imbalance and is not safe for operating.
- It requires an open physical space. Although the potential field-based redirected-walking mechanism is able to avoid static (or moving) obstacles in the physical environment that are detectable by the ranging measurement system, it is still possible for the user to collide with them when they are present. The conditions under which the collisions may occur depend on many factors (such as the user's speed, the values of the adjustable parameters of the mechanism, the available physical space, and the way the user walks) or a combination of factors, and such conditions may not be duplicable. For safety considerations, it is necessary to remove any removable obstacles before operating. For experiments conducted in this dissertation, all the objects were moved toward the boundaries (walls) of the physical environment to offer a more open center space.
- The required size of the available physical space depends on many factors (such as the user's speed, the values of the adjustable parameters of the mechanism, and the way the user walks) and is currently undetermined. However, it is noted that the larger the physical space is, the less possible it is for the user to collide with obstacles. For example, a basketball court-sized environment is more suitable than a regular hallway environment.
- The maximum user speed for safely operating the integrated locomotion interface is currently undetermined. This is also one of various situations under which a quantitative evaluation is difficult to conduct. One similar real-life example is that the maximum speed a motor vehicle can achieve before it crashes into something is difficult to determine because such a value depends on all kinds of factors, and it is unsafe to experiment. Currently, the best way to avoid collision is to specify a limit. While the integrated locomotion interface may be capable of being used for running motions, for it to be safely operated, it is required to limit the user motions to the normal walking speed to avoid unpredictable situations.

Parts of the research conducted in this and previous chapters focused on the development of a locomotion interface. Such a locomotion interface has a similar concept as other existing locomotion interfaces in that it acts as an input generator to control the user's motions in virtual environments. Therefore, the efforts were mainly to achieve higher portability and immersion. The security or privacy issues which are of great concern when accessing the on-line virtual environments were not considered.

E. SUMMARY

This chapter presented a ranging measurement system-based redirected-walking mechanism. Such an improved redirected-walking mechanism utilizes the potential field of range measurements provided by the ranging measurement system and calculates an angular velocity to be injected into virtual environments to steer the user's walking direction away from obstacles in the physical environment.

The main contribution of this chapter is the development of the potential field-based redirected-walking mechanism using a ranging measurement system. It was designed to be integrated with the inertial/magnetic sensor-based locomotion interface for avoiding obstacles in the physical environment when navigating through virtual environments.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION AND RECOMMENDATIONS

This chapter summarizes the major contributions of this dissertation and provides recommendations for future work. The main focus of the research was to develop a locomotion interface that would contribute to self-contained, portable, and immersive VE systems. The developed locomotion interface allows the user to use natural walking motions to navigate through virtual environments. As a means of building such a locomotion interface, the feasibility of utilizing an inertial/magnetic sensor-based system and a ranging measurement system was investigated. Algorithms for the two individual systems and for the integrated system were developed to provide the necessary functionalities of the locomotion interface.

A. CONTRIBUTIONS

The main contribution of this dissertation is the development of a novel locomotion interface using the integration of inertial/magnetic measurement units and a ranging measurement system. Such an integrated locomotion interface contributes to the construction of self-contained, portable, and immersive VE systems that are able to operate in arbitrary open spaces. The specific contributions are summarized as follows.

- A new, real-time, line-based map-building and localization process was developed. Such a process utilizes the information of the physical environment collected by the ranging measurement system to build the 2D line-based map and to estimate the orientation and position of the user within this map. The process was constructed based on the original iterative closest point (ICP) algorithm. New mechanisms including an outlier rejection mechanism and a local minimum escape mechanism were developed to improve the scan-matching result of the ICP algorithm. A new line feature-based transformation correction mechanism was developed for further correcting the residual scan-matching error.
- A new concept of separating the map-building and localization processes was proposed and implemented for the HIVE-like (i.e., simple rectangular-shaped) environments to obtain better map-building and localization results.
- A new locomotion interface was developed using the inertial/magnetic sensor modules. Such a locomotion interface utilizes the motions detected by the inertial/magnetic measurement units attached to the user's head and

foot and relates them to the virtual head and walking motions. It allows the use of natural walking motions to navigate through virtual environments.

- A new redirected-walking mechanism was developed using a ranging measurement system. Such a mechanism can be integrated with the inertial/magnetic sensor-based locomotion interface in contribution to the construction of self-contained, portable, and immersive VE systems. It features a new, potential field-based redirected-walking mechanism used to redirect the user's walking direction away from obstacles (such as walls) in the physical environment when navigating through virtual environments. A relative obstacle avoidance approach is adopted in this mechanism. Such an approach utilizes only the local information of the environment from one single scan of the ranging measurement system at the current moment.

B. RECOMMENDATIONS FOR FUTURE WORK

The presented ranging measurement system-based redirected-walking mechanism utilizes the relative obstacle avoidance approach. Such a mechanism is effective for avoiding obstacles when there is only one user in the physical environment at a time.

For the future development, the use for multiple users in the same physical environment at the same time may be taken into consideration. The relative obstacle avoidance approach that reacts only based on the local information obtained from one single scan of the ranging measurement system may not be sufficient to avoid collision among all the users. There may be a need to adopt the absolute obstacle avoidance approach. Such an approach is to utilize the complete knowledge of the physical environment (such as the global map describing the obstacle/boundary information) and the user's orientation and position within the physical environment. Furthermore, the absolute approach may also need to keep track of the orientation/position information of all the other users for the multi-user application in the same physical environment. Therefore, certain mechanisms may need to be incorporated for sharing such information among all the users.

Although it is supposable that the absolute approach maintaining the complete knowledge regarding the physical environment and all the users may be able to provide proactive collision avoidance, the means to utilize such knowledge for achieving the desired results is still undetermined. Therefore, algorithms may need to be developed for realizing such a concept in the future development of the redirected-walking mechanism.

Further consideration for future work may be to refine the map-building and localization process by investigating different sensing techniques, and determining the scan representations and scan-matching algorithms that best complement the chosen sensing techniques. Such process may contribute to the development of the redirected-walking mechanism using the absolute approach. For other applications, such as the 3D modeling, the map-building and localization process may be modified to utilize a 3D rangefinder (scanner) for building the 3D models of the physical environments.

The locomotion interface may also be further refined by investigating alternative sensor modules to be used and locations to be mounted with the chosen sensor modules. The arm and hand motions may also be incorporated into the locomotion interface for providing users a more immersive experience.

Studies may be conducted to investigate the performance of the developed locomotion interface, which is currently difficult to quantify. Furthermore, the psychological and physiological issues associated with the use of such a locomotion interface may also need to be addressed in the future work.

The recommendations for future development discussed earlier were not meant to be comprehensive. Additional effort may still be required before this dissertation work may be used for actual applications in education and training.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Bob G. Witmer and Michael J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence*, vol. 7, no. 3, June 1998, pp. 225–240.

[2] Shaleph O'Neill, "Presence, place and the virtual spectacle," *PsychNology Journal*, vol. 3, no. 2, 2005, pp. 149–161.

[3] Ioannis Messinis, Dimitrios Saltaouras, Panayotis Pintelas, and T.A. Mikropoulos, "Investigation of the relation between interaction and sense of presence in educational virtual environments," *International Conference on e-Education, e-Business, e-Management and e-Learning*, 2010, pp. 428–431.

[4] Mel Slater, Anthony Steed, John McCarthy, and Francesco Maringelli, "The influence of body movement on subjective presence in virtual environments," *Human Factors*, vol. 40, no. 3, 1998, pp. 469–477.

[5] SENSICS. (2010, Oct.). *xSight and piSight Head Mounted Displays User's Guide*. Columbia, MD.

[6] Carolina Cruz-Neira, Daniel J. Sandin. Tomas A. DeFanti, Robert V. Kenyon, and John C. Hart, "The cave: audio visual experience automatic virtual environment," *Communications of the ACM*, vol. 35, no. 6, pp. 65–72.

[7] Wikimedia Commons, "File:CAVE Crayoland.jpg." [Online]. Accessed Dec. 16, 2013. Available: http://commons.wikimedia.org/wiki/File:CAVE_Crayoland.jpg.

[8] Wikimedia Commons, "File:Vehicle simulator.jpg." [Online]. Accessed Dec. 16, 2013. Available: http://commons.wikimedia.org/wiki/File:Vehicle_simulator.jpg

[9] University of Utah School of Computing, "Locomotion display." [Online]. Accessed Dec. 16, 2013. Available: http://www.cs.utah.edu/research/areas/ve/LocomotionDisplay.html.

[10] Abhijeet Vijayakar and John M. Hollerbach, "A proportional control strategy for realistic turning on linear treadmills," *Proceedings of the 10th Symp. On Haptic Interfaces for Virtual Envir. & Teleoperator Systs.*, 2002, pp. 231–238.

[11] Hiroo Iwata, "The Torus treadmill: realizing locomotion in VEs," *IEEE Computer Graphics and Applications*, Nov/Dec 1999, vol. 19, pp 30–35.

[12] Hiroo Iwata, "Walking about virtual environments on an infinite floor," *Proceedings of the IEEE Virtual Reality*, 1999, pp. 286–293.

[13] Rudolph P. Darken, William R. Cockayne, and David Carmein, "The omni-directional treadmill: A locomotion device for virtual worlds," *Proceedings of UIST*, Banff, Canada, Oct. 14–17, 1997, pp. 213–221.

[14] Laroussi Bouguila and Makoto Sato, "Virtual locomotion system for large-scale virtual environment," *Proceedings of the IEEE Virtual Reality*, 2002, pp. 292–292.

[15] Jia Wang and Robert W. Lindeman, "Comparing isometric and elastic surfboard interfaces for leaning-based travel in 3D virtual environments," *IEEE Symposium on 3D User Interfaces*, Orange County, CA, Mar. 4–5, 2012, pp. 31–38.

[16] NVIS, "nVisor SX60." [Online]. Accessed Dec 18, 2013. Available: http://www.nvisinc.com/product.php?id=16.

[17] InterSense, "InertiaCube2+." [Online]. Accessed Dec 16, 2013. Available: http://www.intersense.com/pages/18/55/.

[18] LORD MicroStrain Sensing System, "3DM-GX3-25." [Online]. Accessed Dec 16, 2013. Available: http://www.microstrain.com/inertial/3DM-GX3-25.

[19] Hokuyo, "UTM-30LX." [Online]. Accessed Dec 16, 2013. Available: http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html.

[20] John Folkesson, John Leonard, Jacques Leederkerken, and Rob Williams, "Feature tracking for underwater navigation using sonar," *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct 29 - Nov 2, 2007, pp. 3678–3684.

[21] Richard J. Rikoski and John J. Leonard, "Trajectory sonar perception," *Proceedings of the 2003 IEEE International Conference on Robotics & Automations*, Taipei, Taiwan, Sept 14–19, 2003, pp.963-970.

[22] Michael Brady, Hugh Durrant-Whyte, Huosheng Hu, John Leonard, Penelope Probert, and B. S. Y. Rao, "Sensor-based control of AGVs," *Computing & Control Engineering Journal*, March, 1990, pp. 64–70.

[23] Chuan Hao Yang, "A person-tracking mobile robot using an ultrasonic positioning system," MS Thesis, Naval Postgraduate School, Dec 2005.

[24] Hokuyo Automatic. (2005, Oct.). *Scanning Laser Range Finder URG-04LX Specifications*. Osaka, Japan.

[25] Hokuyo Automatic. (2011, Nov.). *Scanning Laser Range Finder UTM-30LX-EW Specification.* Osaka, Japan.

[26] MicroStrain. (2009). *Technical Product Overview 3DM-GX3-25 Miniature Attitude Heading Reference System.* Williston, VT.

[27] InterSense. (2001). *InertiaCube2 Bringing 3D To Life Manual for Serial Port Model*. Burlington, MA.

[28] Xiaoping Yun, Eric R. Bachmann, Hyatt Moore IV, and James Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, April 10–14, 2007.

[29] James Calusdian, "A personal navigation system based on inertial and magnetic field measurements," Doctoral dissertation, Naval Postgraduate School, September 2010.

[30] Xiaoping Yun, James Calusdian, Eric Bachmann, and Robert McGhee, "Estimation of human foot motion during normal walking using inertial and magnetic sensor measurements," *IEEE Transactions on Instrumentation and Measurement*, Vol. 61, No. 7, July 2012, pp. 2059–2072.

[31] Xiaoping Yun, Eric Bachmann, and Robert B. McGhee, "A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements," *IEEE Transaction on Instrumentation and Measurement*, Vol. 57, No. 3, March, 2008, pp. 638–650.

[32] John J. Leonard and Hugh F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS*, Osaka, Japan, Nov 3–5, 1991, pp.1442-1447.

[33] Christopher M. Smith, John J. Leonard, Andrew A. Bennett, and Christopher Shaw, "Feature-based concurrent mapping and localization for AUVs," *Proceedings of the MTS/IEEE Conference on Oceans*, vol. 2, 1997, pp. 896–901.

[34] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, June 2006, pp. 99–108.

[35] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics and Automation Magazine*, September 2006, pp. 108–117.

[36] Francesco Amigoni, Simone Gasparini, and Maria Gini, "Building segment-based maps without pose information," *Proceedings of the IEEE*, Vol. 94, No. 7, 2006, pp. 1340–1359.

[37] Francesco Amigoni, Simone Gasparini, and Maria Gini, "Good experimental methodologies for robotic mapping: a proposal," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, April 10–14, 2007, pp. 4176–4181.

[38] Alberto Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, June 1987, pp. 249–265.

[39] Xue-Cheng Lai, Cheong-Yeen Kong, Shuzhi Sam Ge, and Abdullah Al Mamun, "Online map building for autonomous mobile robots by fusing laser and sonar data," *Proceedings of the IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, July 2005, pp. 993–996.

[40] Edwin Olson, John Leonard, and Seth Teller, "Fast iterative alignment of pose graphs with poor initial estimates," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, May, 2006, pp. 2262–2269.

[41] Jack Collier, and Alejandro Ramirez-Serrano, "Environment classification for indoor/outdoor robotic mapping," *Proceedings of the 2009 Canadian Conference on Computer and Robot Vision*, 2009, pp. 276–283.

[42] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton, "Redirected walking," *Eurographics*, 2001.

[43] Sharif Razzaque, David Swapp, Mel Slater, Mary C. Whitton, and Anthony Steed, "Redirected walking in place," *Eight Eurographics Workshop on Virtual Environments*, 2002, pp. 123–129.

[44] Sharif Razzaque, "Redirected walking," Doctoral dissertation, University of North Carolina, Chapel Hill, 2005.

[45] Tabitha C. Peck, Mary C. Whitton, and Henry Fuchs, "Evaluation of reorientation techniques for walking in large virtual environments," *IEEE Conference on Virtual Reality*, Reno, Nevada, March 8–12, 2008, pp. 121–127.

[46] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton, "Evaluation of reorientation techniques and distractors for walking in large virtual environments," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 3, May/Jun 2009, pp. 383–394.

[47] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton, "Improved redirection with distractors: a large-scale-real-walking locomotion interface and its effect on navigation in virtual environments," *IEEE Conference on Virtual Reality*, Waltham, Massachusetts, March 20–24, 2010, pp.35-38.

[48] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton, "The design and evaluation of a large-scale real-walking locomotion interface," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 18, No. 7, July, 2012, pp. 1053–1067.

[49] Radu Gabriel Danescu, "Obstacle detection using dynamic particle-based occupancy grids," *IEEE International Conference on Digital Image Computing Techniques and Applications*, 2011, pp.585-590.

[50] Marcus Konrad, Dominik Nuss, and Klaus Dietmayer, "Localization in digital maps for road course estimation using grid maps," *IEEE Intelligent Vehicles Symposium*, Alcala de Henares, Spain, June 3–7, 2012, pp. 87–92.

[51] Paul J. Besl and Neil D. McKay, "A method for registration of 3-D shapes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, February 1992.

[52] Haili Chui and Anand Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding - Special issue on nonrigid image registration*, Volume 89 Issue 2–3, pp. 114–141, February 2003.

[53] Berthold K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, Vol. 4, pp. 629, April 1987.

[54] Jeff M. Phillips, Ran Liu, and Carlo Tomasi, "Outlier robust ICP for minimizing fractional RMSD," *Proceedings of the IEEE International Conference on 3-D Digital Imaging and Modeling*, 2007, pp. 427–434.

[55] Szymon Rusinkiewicz and Marc Levoy, "Efficient variants of the ICP algorithm," *Proceedings of the IEEE International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.

[56] S. May, D. Droeschel, D. Holz, S. Fuchs, and A. Nuchter, "Robust 3D-mapping with time-of-flight cameras," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Missouri, USA, 2009, pp. 1673–1678.

[57] Dirk Holz and Sven Behnke, "Sancta simplicitas–on the efficiency and achievable results of SLAM using ICP-based incremental registration," *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 3–8, 2010, pp. 1380–1387.

[58] Songrit Maneewongvatana and David M. Mount, "It's ok to be skinny, if your friends are fat," *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, 1999.

[59] Mark A. Haidekker, " The Hough transform," in *Advanced Biomedical Image Analysis*, John Wiley & Sons, 2011, ch. 7, pp. 211–235.

[60] Xiaoping Yun, Khine Latt and J.Scott Glennon, "Mobile robot localization using the Hough transform," *Proceeding of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference Gaithersburg*, MD, September 14–17, 1998.

[61] David Waller, Eric Bachmann, and Eric Hodgson, "The HIVE: A huge immersive virtual environment for research in spatial cognition," *Behavior Research Methods*, Nov 2007, 39 (4), pp. 835–843.

[62] Stuart J. Russell and Peter Norvig, "Hill-climbing search," in *Artificial Intelligence: A Modern Approach*, 3rd ed., New Jersey, Pearson Education, 2010, ch. 4, sec. 1, pp. 122–125.

[63] Stuart J. Russell and Peter Norvig, "Simulated annealing," in *Artificial Intelligence: A Modern Approach*, 3rd ed., New Jersey, Pearson Education, 2010, ch. 4, sec. 1, pp. 125.

[64] Wikipedia, "First-person (video games)." [Online]. Accessed Dec 16, 2013. Available: https://en.wikipedia.org/wiki/First-person_(video_games).

[65] Nintendo, "Wii." [Online]. Accessed Dec 16, 2013. Available: http://www.nintendo.com/wii/what-is-wii/.

[66] Microsoft, "Kinect for XBox 360." [Online]. Accessed Dec 16, 2013. Available: http://www.xbox.com/en-US/xbox360/accessories/kinect/KinectForXbox360.

[67] Microsoft, "KINECT for Windows." [Online]. Accessed Dec 16, 2013. Available: http://www.microsoft.com/en-us/kinectforwindows/.

[68] John Hollerbach, David Grow, and Craig Parker, "Developments in locomotion interfaces," *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*, Chicago, IL, June 28 - July 1, 2005, pp. 522–525.

[69] Murat Onder, “Locomotion in virtual environments and analysis of a new virtual walking device,” MS Thesis, Naval Postgraduate School, Mar 2005.

[70] Mary Whitton, “Getting from here to there: locomotion in virtual environments,” *2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications*, 2010.

[71] Michael A. Goodrich, “Potential field tutorial,” Lecture Notes, Brigham Young University, Provo, UT, 2002.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California